Finite Elements in Analysis and Design 51 (2012) 31–48

Contents lists available at SciVerse ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

General purpose software for efficient uncertainty management of large finite element models

Edoardo Patelli*, H. Murat Panayirci, Matteo Broggi, Barbara Goller, Pierre Beaurepaire, Helmut J. Pradlwarter, Gerhart I. Schuëller [1]

Engineering Mechanics, University of Innsbruck, Technikerstraße 13, 6020 Innsbruck, Austria

ARTICLE INFO

Article history:
Received 3 December 2010
Received in revised form
15 November 2011
Accepted 15 November 2011
Available online 14 December 2011

Keywords:
Uncertainty quantification
Surrogate model
Finite element analysis
General purpose software
Sensitivity analysis
Optimization
Reliability
Stochastic finite element methods

ABSTRACT

The aim of this paper is to demonstrate that stochastic analyses can be performed on large and complex models within affordable costs. Stochastic analyses offer a much more realistic approach for analysis and design of components and systems although generally computationally demanding. Hence, resorting to efficient approaches and high performance computing is required in order to reduce the execution time.

A general purpose software that provides an integration between deterministic solvers (i.e. finite element solvers), efficient algorithms for uncertainty management and high performance computing is presented. The software is intended for a wide range of applications, which includes optimization analysis, life-cycle management, reliability and risk analysis, fatigue and fractures simulation, robust design.

The applicability of the proposed tools for practical applications is demonstrated by means of a number of case studies of industrial interest involving detailed models.

1. Introduction

In many engineering fields, computational approaches are used to characterize, predict, and simulate complex systems. The problems addressed by numerical analysis face continuously increasing demands, both in terms of efficiency and accuracy.

Within the community of computational mechanics, currently mostly deterministic analyses are performed [40]; nonetheless, such deterministic analyses provide insufficient information to capture the variability of the e.g. structural response due to the inevitable uncertainties in loading, materials and manufacturing quality. The software to perform deterministic analyses are generally quite sophisticated in terms of geometrical and mechanical modeling capabilities, continuously improved and updated to account for new trends and developments in computational mechanics.

Although stochastic methods offer a much more realistic approach for analysis and design of systems, the utilization of such tools in practical applications remains quite limited. One of the reasons to explain this fact is that the developments of general purpose software for stochastic analysis have received considerable less attention than their deterministic counterparts. Another common limitation of stochastic procedures is that the computational cost is often by orders of magnitude higher than the deterministic analysis. This is because in general, instead of running the analysis code only once, the stochastic analysis involves many repeated executions. Therefore, performing uncertainty analysis might lead to impractical computational costs especially for detailed models.

Within this context, the aim of this contribution is presenting and discussing efficient approaches for stochastic analysis and their applications to large FE models for the solution of realistic engineering problems. This aim is achieved by resorting to a general purpose software developed at the Institute of Engineering Mechanics, University of Innsbruck, Austria. Originally developed for performing only stochastic structural analysis, the current version of the software is intended for a wide range of applications, which includes optimization analysis, life-cycle management, reliability and risk analysis, fatigue and fractures simulation, robust design.

The outline of the present paper is as follows: Section 1.1 reports a brief overview of the computationally challenging stochastic problems of interest in engineering practice. In Section 2, a general purpose software intended for solving efficiently a large range of stochastic problems is described. In Section 3, a number of case studies are presented to demonstrate the applicability of the software for solving problems of practical interest. Finally, some final remarks are listed in Section 4.

* Corresponding author. Tel.: +44 151 7944079; fax: +44 151 7944703.
E-mail addresses: edoardo.patelli@liverpool.ac.uk (E. Patelli), mechanik@uibk.ac.at (G.I. Schuëller).
URL: http://mechanik.uibk.ac.at (G.I. Schuëller).
[1] Tel.: +43 512 507 6841; fax: +43 512 507 2905.

0168-874X
doi:10.1016/j.finel.2011.11.003

1.1. Importance of stochastic analysis

Knowledge about the future behavior of structures, systems and components is the basis for far reaching economical and safety relevant decisions in our society and appears on different engineering fields (e.g. automotive industry, aerospace, mechanical and civil engineering, etc.). Together with observed responses, it provides the basis to broaden the mechanical and physical understanding between action and reaction. In order to predict accurately the behavior of systems and structures, mathematical models must be constructed and then evaluated. The FE methods and the exponential growth of the computational power allow to solve partial differential equations, i.e. mathematical models, for realistic boundary conditions with high accuracy.

Comparing response predictions with measured data, however, could not show that the fidelity has improved as much as the abilities of detailed modeling and of accurate analysis. The reasons for this discrepancy are twofold: parameter and model uncertainty. Parameter uncertainties denote input data in the computational model which are not precisely known and must be expected to deviate from the assumed deterministic values. Model uncertainties denote the fidelity of the mathematical model, which involves usually some abstraction and simplifying assumptions, to represent with sufficient accuracy the actual mechanical/physical response. It is nowadays widely recognized that essential progress in virtual response prediction can only be accomplished in case uncertainties are included in the analysis, see e.g. [22].

The most widely accepted method to deal rationally with uncertainties is the stochastic approach including Bayes and Laplace's subjective interpretation of probability as a state of information. Alternative approaches are valid options, e.g. fuzzy logic [25] and possibility theory, which are not so far developed as the theory of probability. In the stochastic approaches, uncertainties are represented mathematically by random quantities and by suitable probability distributions. The stochastic analysis allows for Uncertainty Quantification (UQ) and its propagation to the responses, which are mathematically seen as random quantities completely specified by their probability distribution.

The merits of considering uncertainties are manifold: it allows for assessing the reliability and variability of the response(s) and, most importantly, it provides information to increase the reliability and to improve the design as well as the fidelity of the prediction. Sensitivity analyses reveal the quantities which are mainly responsible for the variability of the response(s). In case the uncertainty is due to the lack of knowledge (epistemic type) and therefore reducible, the fidelity of the prediction can be improved by gathering additional data for those quantities which cause most uncertainty in the response(s). Irreducible (aleatory) uncertainties lead to irreducible uncertainties in the response. In this case the design must be such that adverse events do not jeopardize safe operation. The quantification of the tolerable failure probability allows for an economic investment against failure.

Similarly, uncertainty quantification and propagation are important aspects when trying to optimize a system or a component. Optimal solutions in a deterministic setting might not perform as expected and can be even dangerous in case ignored uncertainties influence the performance considerably. On the other hand, robust design procedures take into account all relevant uncertainties and provides robust and sound solutions, e.g. the failure probability is constrained to be less than an acceptable value. Furthermore, decision within life cycle management for important infrastructures and investments must be done based on incomplete and generally insufficient data, where a probabilistic Bayesian approach could provide valuable information.

2. Efficient implementations and strategies for stochastic analysis

2.1. General remarks

The term *general purpose software* means that a reasonably wide range of engineering and scientific problems can be treated by a single software. This is in contrast with specialized software, which are developed for solving only a specific type of problem within a particular discipline. In general, the capabilities of the *specialized software*, optimized for solving problems for which they are designed for, are not completely covered by general purpose software. However, general purpose software are much more flexible and their advantages for the user are manifold, mainly the possibility to solve different types of problems adopting a single software. This results in a drastic reduction in the efforts that an analyst needs to invest to familiarize with the software. Furthermore, general purpose software offer the possibility to customize the solution sequences becoming able to solve problems that were not even prefigured during the design of the software.

The downside of general purpose software packages is that, usually, they are much more complex than dedicated software in terms of number of lines of code required, structure and time required for developing and testing. However, since general purpose software are designed to solve a broad variety of problems, they are usually also much simpler to use and they are not only to be adopted by skilled users.

An overview of software to model uncertainties in loads, material properties and geometries for design and decision-making purposes is available e.g. in [39].

2.2. Structure of the software

The structure of general purpose software is generally composed of three main blocks: user interfaces, core components and a set of tools for the interaction with external codes (i.e. 3rd party software). Each of these main blocks can be composed of a number of additional sub-components. The scheme of the general purpose software as considered in this paper is shown in Fig. 1.

The general purpose software provides a large set of different tools and methods that can be combined to solve a specific problem.

The software core components are coded in a object oriented fashion in MATLAB® environment, which provides an expandable modular framework.

2.3. User interfaces

General purpose software offers in general different ways to interact with it. The most common way is by means of user friendly Graphical User Interfaces (GUI). These interfaces are intended to be used to solve all kind of problems supported by the software. The GUI of the software considered in this work, shown in Fig. 2, is coded in Eclipse RCP, a framework that allows to deploy native GUI applications to a variety of desktop operating systems, such as Windows, Linux and Mac OS X. The general purpose GUI provides wizards and guides to assist inexperienced users to set up the problem and to select the most appropriate tools required by the analysis. Furthermore, it includes a very powerful input/output editor regarding the interaction with 3rd-party software as shown in Section 2.4.

Also, a command line interface based on Matlab scripting, shown in Fig. 3, provides a high-level powerful and flexible programming environment, which allows advanced users and researchers to modify pre-written solution sequences, explore data, define algorithms, and

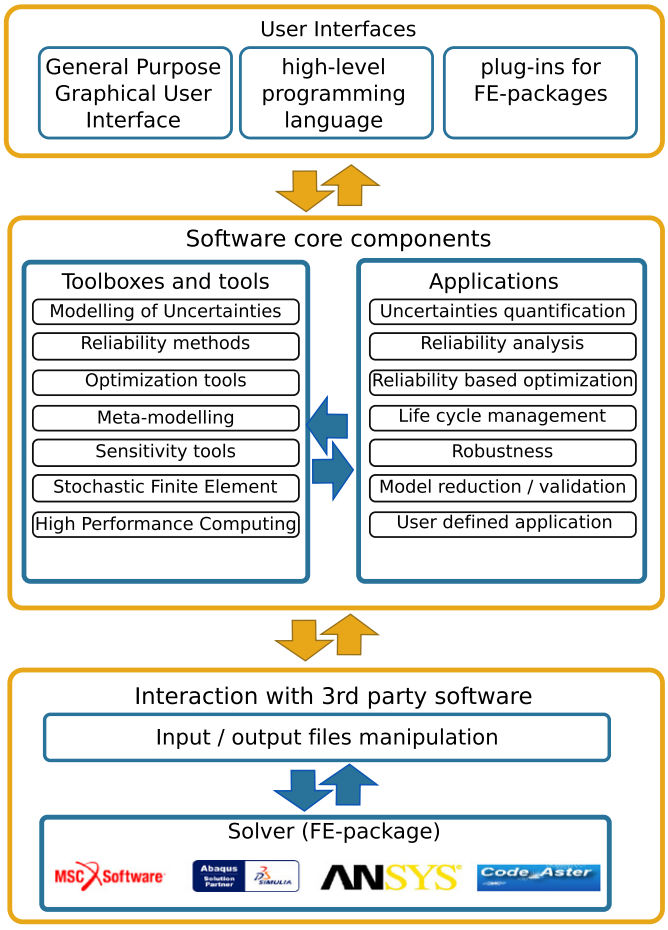

Fig. 1. Schematic representation of general purpose software for computational stochastic analysis.

create custom tools that provide early insights and competitive advantages to solve specialized problems.

Finally, user interfaces provided as plug-ins (i.e. extensions) for commercial pre- and post-processor software, e.g. Patran®, are also available, as shown in Fig. 4. Thanks to these plug-ins, the user can define stochastic analyses directly on the GUI of the pre- and post-processor software, which the user is already familiar with.

2.4. Interaction with 3rd party software

In general, an analyst is able to solve a deterministic problem using a specific solver, for instance a commercial FE-solver or an in-house solver. Consequently it would be advantageous to offer the possibility to perform stochastic analysis adopting the same solver. In order to achieve this, it is necessary to resort to a series of different communication tools to interact with standard solvers used in industry, i.e. FE packages such as Nastran, Ansys, Abaqus, FEAP, code_Aster, etc.

From the point of interaction with 3rd party software, the methods can be classified into two groups, namely intrusive and non-intrusive methods, respectively. On one hand, non-intrusive methods refer to those, where 3rd party programs can be used in a black-box fashion, i.e. no modification is required on the 3rd party program, see e.g. [8]. The intrusive methods, on the other hand, require calculations with system matrices, see e.g. [6]. Therefore, these matrices have to be accessed in the deterministic solver. This may require advanced knowledge or even modifications on the solver side. Consequently, the implementation of this class of methods is more involved.

Non-intrusive methods: In this class of methods, the interaction is based on the manipulation of the input/output files without modifications in the 3rd party solver. The basic concept is to use the original input files, in ASCII format, as a basis and to modify these files for generating valid input configurations automatically for the numerical analysis. For this purpose, the quantities associated with uncertainty and/or design variables are replaced in the original ASCII

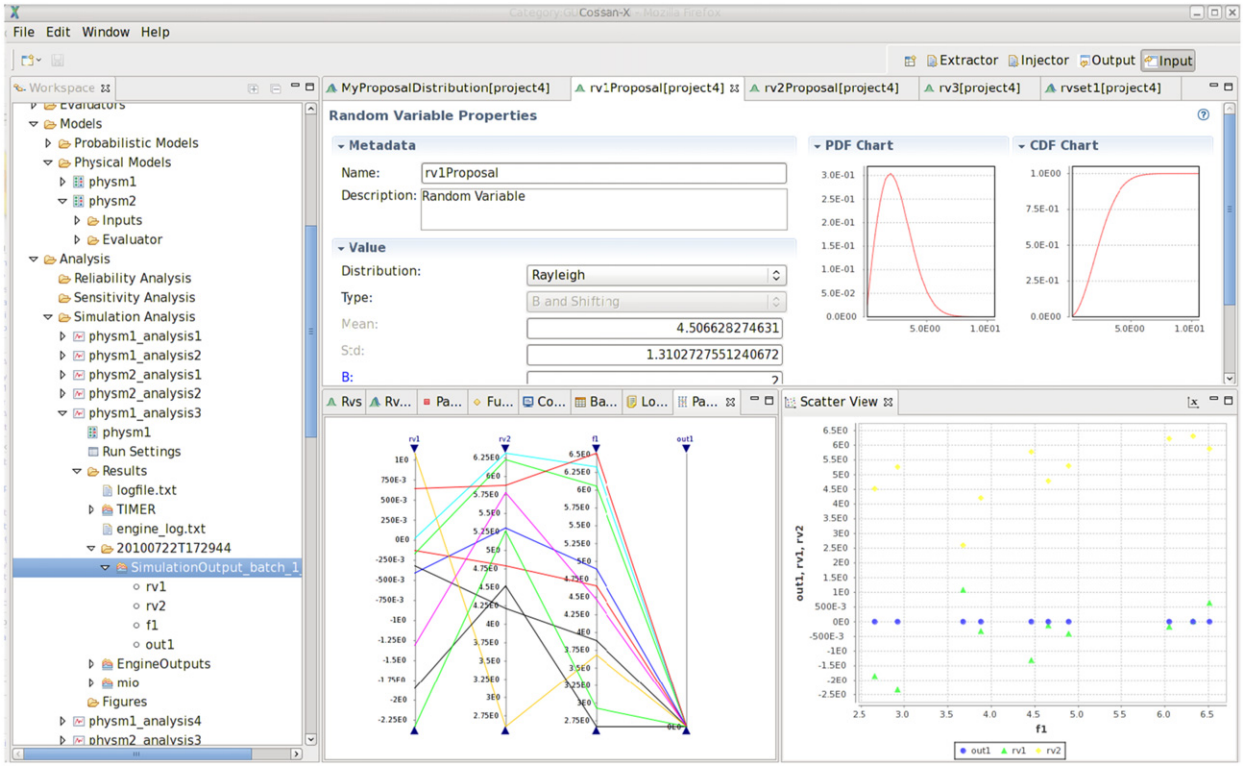

Fig. 2. General purpose graphical user interfaces.

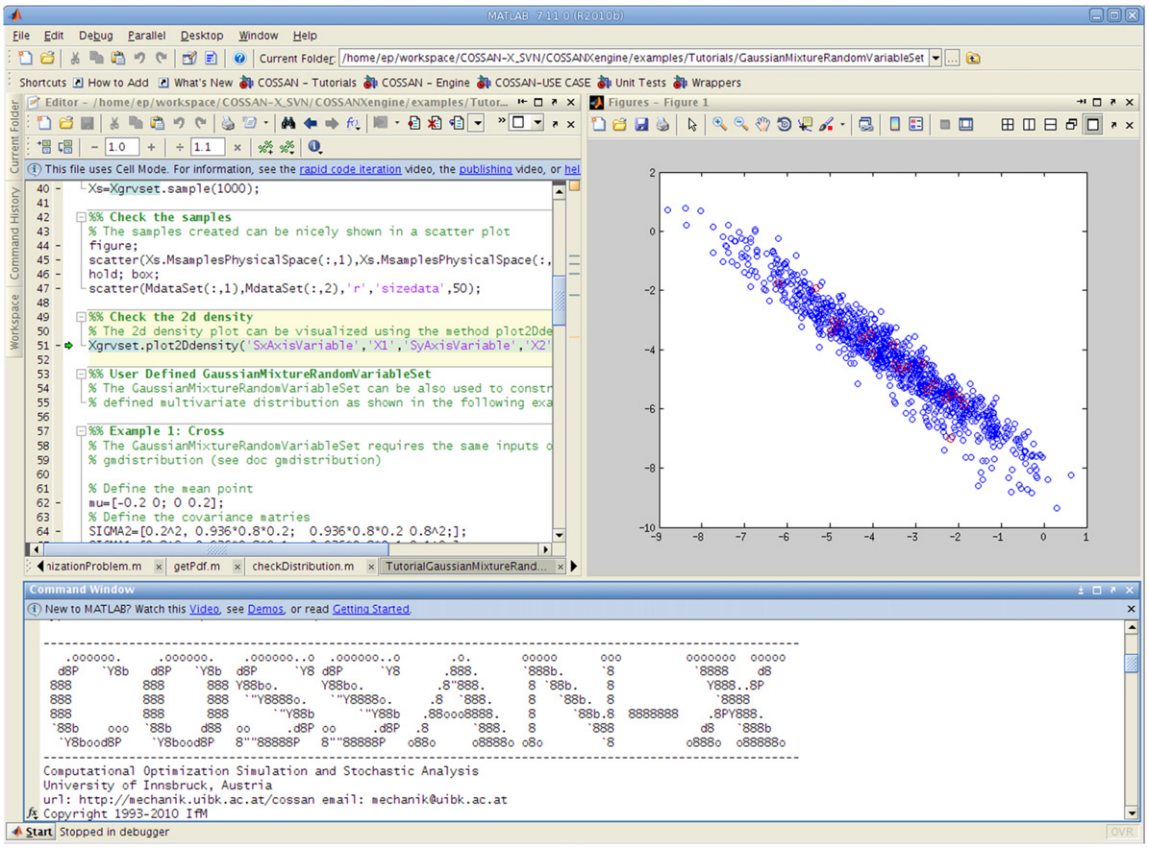

Fig. 3. Matlab command line interface.

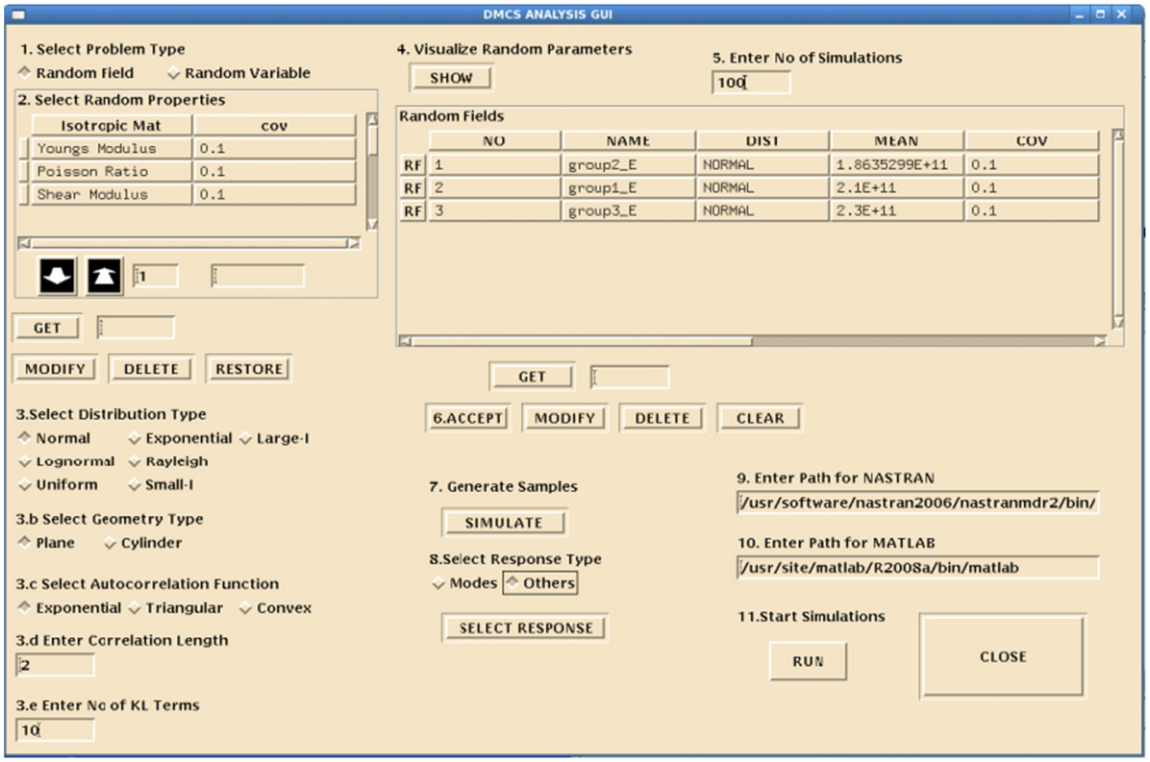

Fig. 4. Patran plug-in for SFEM analysis.

input files by some identifiers, i.e. strings of unique characters. These identifiers establish the relation between the variables defined in general purpose software and the physical variables of the model.

Regarding the extraction of the quantity of interest from the output files of the 3rd party software, different post-processor tools are offered. In fact, all the FE-solvers are able to export the

quantity of interest in ASCII output files. Using these output files, the relative and/or absolute positions of the quantities of interest are defined and successively used to extract the values of interest.

The manipulation of the input files and the extraction of the quantities of interest from the output files is completely automatized in the software considered here.

Intrusive methods: Intrusive methods are those where the data structures in the 3rd party software have to be accessed.

The levels of communication between stochastic and deterministic solvers are especially important in this regard. For example, if the transfer of system matrices rely on file input/output between these software, the overhead associated with this data transfer becomes quite significant for large systems. Furthermore, it should be noted that the required data vary according to the formulation of the considered approach. Based on this, it can be said that in case the implementation can access the data structures and modify the program flow of the FE code, improved performance may be accomplished. These conditions are clearly met in context with open-source FE codes. However, also proprietary software may provide the means for accessing data structures such as stiffness matrices and utilize those in an efficient implementation of various methods.

Considering the various ways of interaction with respect to intrusive methods, the most straightforward approach for the transfer of system matrices from the 3rd party solvers is using files. There are various parameters, which affect the computational efficiency of this step:

1. *File format:* Different file formats are offered by different FE solvers to input/output (I/O) matrices. For example, MSC.Nastran provides file formats such as *op2* and *op4*, whereas in ANSYS the Harwell–Boeing format is available for the same purpose. The format *op4* refers to ASCII files, which are readable on any conventional editor, and *op2* is a FORTRAN binary option, which requires a FORTRAN program to provide readable results. The number of digits retained in each value in these files is controlled by an input parameter of the corresponding function call. The FORTRAN binary option file is typically one-third the size of the ASCII version.
2. *Matrix storage format:* FE models of complex structures may result in large system matrices, which have usually many zero entries. Hence their format, i.e. sparse or non-sparse, can become decisive on the computational efficiency, as well as on handling the calculations from the memory allocation point of view. Most of the FE solvers provide the option to output the system matrices in sparse format.
3. *Code to I/O the matrix:* The libraries which are used for I/O matrices via files, influence the computational efficiency of the implementation significantly and hence should be optimized for high efficiency.

2.5. Toolboxes (analysis types)

An approach of using various layers has been considered within the implementation of the methods and features. The toolboxes-layer represents the core components of the software and implements the state of the art in stochastic analysis that have been shown to represent a robust and efficient approach for the uncertainty management (see e.g. [42]). The combination of various algorithms with specific solution sequences permits the analysis of engineering problems as shown by different applications presented in Section 3. These algorithms then eventually form the applications-layer, such as UQ, reliability analysis, life cycle management, etc.

2.5.1. Modeling of the uncertainties

Uncertainties can be described within the framework of probability. Scalar values can be modeled using random variables (RVs), e.g. static load; time variant quantities can be represented using stochastic processes, e.g., wind speed or earthquake excitation; space variant quantities can be described using random fields, e.g. material properties in a solid.

In the software described in this work, a RV is defined by specifying the distribution type, e.g. normal, log-normal, uniform, etc., together with either the parameters of the distribution, or its moment(s). Alternatively, a RV can be constructed starting from a set of realizations. In the latter case, the parameters leading to an optimal fit of the set of realizations by a specific distribution are automatically determined using the maximum likelihood method, which is very versatile tool and yields estimators of the distribution parameter with optimal statistical properties (see e.g. [24]).

The software allows to define multivariate distributions by defining the marginal distributions and introducing a correlation matrix between them. Samples from multiple correlated distributions cannot be generated directly, since pseudo-random number generators are supposed to generate independent samples. Thus, in order to generate samples from correlated multivariate distributions, it is necessary to transform the random variable space to obtain an uncorrelated multivariate distribution, where independent RVs can be used. This requirement is fulfilled in the so-called *standard normal space*, which is an uncorrelated multi-dimensional random variable space with zero mean, unit standard deviation Gaussian marginal probability density functions.

Let us consider a set of RVs $(X_1,\ldots,X_n)$ having correlation coefficients ρ_{ij}, $i,j=1,\ldots,n$ and marginal cumulative density functions F_{X_i}, $i=1,\ldots,n$. First, a transformation to correlated standard normal variables is performed according to

$$z_i = \Phi^{-1}(F_{X_i}(x_i)) \quad \text{for } i=1,\ldots,n \tag{1}$$

where Φ^{-1} denotes the inverse of the standard normal cumulative density function, i.e. cumulative density function of a Gaussian distribution with zero-mean and unit variance. After the transformation described in Eq. (1), the correlation coefficient ρ'_{ij} of the variables is so that Eq. (2) is verified [18]

$$\rho'_{ij} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(\frac{x_i-\mu_i}{\sigma_i}\right)\left(\frac{x_j-\mu_j}{\sigma_j}\right)\varphi(z_i,z_j,\rho_{ij})\,dz_i\,dz_j \tag{2}$$

where φ denotes the probability density function of a bivariate standard normal Gaussian distribution. An analytic approximation of ρ'_{ij} is adopted if applicable [18], alternatively, a numerical solution of Eq. (2) is carried out using Monte Carlo (MC) simulation. The RVs in the standard normal space $(u_1,\ldots,u_n)$ are expressed as

$$u_i = \sum_{j=1}^{n} b_{ij} z_j \quad \text{for } i=1,\ldots,n \tag{3}$$

where b_{ij} denotes the terms of the inverse of the Choelsky decomposition of the correlation matrix defined by the terms ρ'_{ij}. Finally, isoprobabilistic transformation to the physical space of the correlated distributions is carried out.

Multivariate distributions can also be defined using limited experimental data, adopting an efficient approach based on a mixture of one or more multivariate Gaussian distribution components [12]. In this method, the probability density function of such distribution is defined as the sum of n kernels with a Gaussian distribution

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{i=1}^{n} a_i f_{\boldsymbol{Y}_i}(\boldsymbol{x}) \tag{4}$$

where $f_{\boldsymbol{X}}$ denotes the multivariate probability density function of a multidimensional random variable $\boldsymbol{X}$ defined as a combination

of Gaussian distributions, and $\boldsymbol{x}$ is a point of the sample space. $f_{\boldsymbol{Y}_i}$ indicates the correlated multidimensional Gaussian distributions, centered around support points that can be manually specified or automatically determined in the case experimental realizations are supplied. Finally, a_i, $i=1,\ldots,n$ are used to weigh the contribution of each Gaussian probability density function. The weights are set so that $\sum_{i=1}^{n} a_i = 1$.

Stochastic processes and random fields allow to model quantities showing both a random variability, and a functional dependence in a multidimensional continuous space, i.e., time dependence or spatial variability respectively, (see e.g. [37,51]). In the case the stochastic process (random field) is Gaussian, it is fully defined by the mean function and the covariance function. The covariance function represents the mutual influence of the process at two different spatial-coordinates/time-instants.

For practical engineering applications, the functional dependence is studied for a finite set of discrete coordinates (e.g. time discretization, mesh of a FE model, etc.). In this case, the mean and covariance functions degenerate to a mean vector and covariance matrix, respectively. Realizations from any Gaussian distributed stochastic process (random field) are computed using the Karhunen–Loève expansion [14,19]. For the case of discrete coordinates is

$$Y = \mu_Y + \sum_{i=1}^{n} \sqrt{\lambda_i}\xi_i \Phi_i \tag{5}$$

where the λ_i and Φ_i, $i=1,\ldots,n$ are the eigenvalues and eigenvectors of the covariance matrix, respectively; $\boldsymbol{\xi} = \xi_1,\ldots,\xi_n$ are random values generated from a set of n independent standard normal RVs and μ_Y is the mean vector of the stochastic process (random field). The Karhunen–Loève expansion allows accurate representation for Gaussian processes with a truncation of the sum to $m < n$, using the m biggest eigenvalues of the correlation matrix. The number of terms of the expansion is determined according to the decay of the eigenvalues, so that an acceptable fraction of the variance is captured. The general purpose software allows to generate samples from Gaussian processes only.

2.5.2. Reliability

Reliability engineering provides the theoretical and practical tools whereby the probability and capability of parts, components, equipment, products and systems to perform their required functions can be specified, designed in, predicted, tested and demonstrated [15]. In reliability analysis the quantity of interest is the time to failure, T (often called mission time) and the probability of failure, p_f, for dynamic and static systems, respectively.

For time-invariant problems the evaluation of the p_f of components and systems involves the calculation of the following multidimensional volume integral:

$$p_f = P[F] = \int \cdots \int_F f(\theta_1,\ldots,\theta_n) d\theta_1,\ldots,d\theta_n \tag{6}$$

where F defines the event that causes the failure of the system. This amount is traditionally defined by a so-called scalar performance function $g(\boldsymbol{\theta})$ in a n-dimensional space, where $g(\boldsymbol{\theta}) \le 0$ defines the failure domain F and $g(\boldsymbol{\theta}) > 0$ the safe domain.

Mechanical systems arising in typical engineering applications usually exhibit a degree of complexity that prevents an analytical solution of the failure probability since $g(\boldsymbol{\theta})$ is not known explicitly. The failure probability can, in theory, always be estimated via MC simulation. However, for most practical applications, the evaluation of a single point of the function $g(\boldsymbol{\theta})$ requires a full analysis of the model. Therefore, the computational time required by the MC simulation might become infeasible for the cases, where a substantial number of samples are required by the analysis.

In order to circumvent this drawback, the failure probability may be estimated by means of approximate methods (see e.g. [9]) or advanced simulation-based methods (see e.g. [43]). First and second order reliability method and estimation of bounds can be mentioned within the first category. The family of the advanced simulation-based methods includes importance sampling [44], line sampling [43] and subset simulation [5].

Furthermore, the reliability analysis involves the estimate of the probability of occurrence of a single failure model as well as the probability of occurrence of certain combinations of events, i.e. system reliability (see e.g. [32]).

2.5.3. Optimization

The optimization problem can be defined mathematically as the identification of an optimal solution $\boldsymbol{x}$ of the function g, that is, a feasible solution $f(\boldsymbol{x})$

$$min\{g(x') | x' \in f(x)\} \tag{7}$$

The field of optimization is vast. Therefore, several ways exist to classify the available algorithms (see e.g. [3]). Among these possibilities, a classical criterion is to distinguish between gradient-based and gradient-free algorithms. The reason for selecting this classification approach is due to the fact that the computation of gradients may be rather involved, especially when the functions are noisy. However, gradient-free algorithms usually require much more function evaluations than gradient-based algorithms although they are better to handle noisy functions. Moreover, some of the gradient-free algorithms are also capable of performing global optimization. Therefore, the classification of gradient-based and gradient-free optimization algorithms reflects the trade-off that exists between these two types of methods.

The optimization toolbox provides a set of widely used algorithms to solve constrained and unconstrained, continuous and discrete, standard and large-scale optimization problems. The optimization toolbox includes the gradient-based optimization method Sequential Quadratic Programming (see e.g. [3]) and the following gradient-free algorithms: Genetic Algorithms, Simulated Annealing, Constrained Optimization by Linear Approximation (COBYLA), Cross Entropy and Evolution Strategies (see e.g. [13]).

Combining optimization and reliability methods, it is possible to perform the so-called reliability based optimization analysis and robust design optimization [41], which aims to identify optimal design solutions considering uncertainties and consequently the risk. When such kind of analyses are an integral part of the development cycle to make the end-product, i.e. the component/system, less sensitive to factors that could adversely affect the performance.

2.5.4. Meta-modeling

In case of multiple evaluation of the numerical model, the computational efforts might become infeasible. One way to reduce the analysis time is to use meta-models, which approximate the quantity of interest at low computational cost. Meta-models mimic the behavior of the original model, by creating input–output relations that approximate the real one by means of basic mathematical operations. Surrogate models exist to approximate generic input–output relations, like response surfaces [23] and Artificial Neural Networks (ANN) [1], or can be specifically tailored for particular applications, like the non-probabilistic model [46] and the mode-based meta-model [33] used in structural dynamics.

As an example of application, a meta-model can be used to evaluate an objective function and/or constraints for each trial candidate solution explored within an optimization algorithm. Often, the quantities of interest are retrieved from a time-

consuming analysis, e.g. a detailed FE model. In reliability-based optimization this will be even more demanding, since it requires a full stochastic analysis to compute the probability of failure and the corresponding performance.

Meta-models require to be *calibrated* (also referred to as *trained*) in order to accurately predict the model outputs. This is achieved by feeding a collection of input–output data-sets of the "real"-model only. The inner parameters of the surrogate model are then modified by dedicated calibration algorithms, until a desired accuracy of the meta-model is achieved. An important indicator of the goodness of a meta-model after the training is the coefficient of determination R^2, defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{data}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{data}} (y_i - \overline{y}_i)^2} \tag{8}$$

where y_i are the outputs of the physical or full model, $\overline{y}_i = (1/N_{data}) \sum_{i=1}^{N_{data}} y_i$ and y_i are the outputs predicted by the meta-model. The accuracy of the output prediction of the meta-model can be judged by the closeness of the value R^2 to the target value of 1.0, which expresses an exact match of the surrogate model prediction and the output of the full model. This quantity is computed both using the calibration set and a new set of input/output values, not used by the calibration procedure, called validation or verification set. In this second case, an indication of the generalization capabilities of the meta-model is obtained.

2.5.5. Sensitivity

Sensitivity toolbox allows to study the relationships between the input and output factors, and to identify the most significant variables affecting the results of the model [31,35]. In particular, sensitivity analysis can also be used for model calibration, model validation, decision making process, i.e. any process where it is crucial to identify which are the variables that mostly contribute to the output variability.

Sensitivity analysis may be divided into three broad categories: local sensitivity analysis, screening methods and global sensitivity analysis. Since the sensitivity analysis may be computationally very demanding, the local sensitivity analysis is the only analysis performed in practical cases [36].

Local sensitivity analysis provide information about the system behavior only around a reference point. The local response of the model, obtained by varying input factors One-At-a-Time, is investigated while holding the others fixed to a central (nominal) value. This involves the estimation of the partial derivatives, possibly normalized by the nominal value of the factor or by its standard deviation [7].

The simplest sensitivity measure, $S_i^{(l)}$, is based on the derivative of the function

$$S_i^{(l)} = \frac{\partial Y}{\partial X_i} \tag{9}$$

where X_i represents the uncertain input and Y the quantity of interest considered (e.g. the response of the model). Taking into account only the derivative of the function, this measure is not very informative. An available practice is the normalisation of the derivatives by the nominal values or, more often, by the standard deviations, σ_{X_i}, of the uncertain input, X_i, and normalized by the standard deviation, σ_Y, of the response Y, i.e.

$$S_i^{(\sigma)} = \frac{\sigma_{X_i}}{\sigma_Y} \frac{\partial Y}{\partial X_i} = \frac{\sigma_{X_i}}{\sigma_Y} S_i^{(l)} \tag{10}$$

The local sensitivity analysis (or point sensitivity analysis) provides a quantification for the importance of input variables at a single point in the input domain. Clearly the results of sensitivity analysis depends on the choice of the reference point, except for linear functions.

The sensitivity toolbox includes a MC procedure to estimate the gradient of a function with sufficient accuracy and it has been shown that the procedure is particularly efficient for high dimensional problems (see [34]).

In this MC approach, the gradient of the function, $\nabla g(\boldsymbol{x})$, is not determined directly, as it is the case for finite difference method or direct differentiation procedures. Instead, it is estimated by random sampling in the close neighborhood of an arbitrary reference point, $\boldsymbol{x} = \tilde{\boldsymbol{x}}$.

The global sensitivity analysis techniques take into account the entire range of the variation of the input parameters. Their aim is to apportion the whole output uncertainty according to the different sources of uncertainty in the model inputs (see e.g [35]). Generally, the global sensitivity methods are based on estimating the fractional contribution of each input factor with respect to the total variance of the model under investigation, also called ANOVA (ANalysis Of VAriance) techniques. The so-called Sobol' measure [45] is mostly used in this regard, where each effect is computed by evaluating a multidimensional integral via MC simulation.

Although the sensitivity toolbox provides the most efficient algorithms to perform sensitivity analysis (see e.g. [31,50]), the global sensitivity analysis remains as a powerful but computationally expensive method, especially considering complex industrial applications. For instance, the total number of sensitivity indices that should be estimated is as high as 2^k+1, where k represents the number of uncertain parameters. The resulting cost of the estimate is then $N_t = N(2^k+1)$, where N represents the sample size needed to estimate a single index. It is well known that, for complex computer models, an accurate estimation of these indices by the simple MC method requires $N > 1000$. The computational time can be significantly reduced by replacing the complex computer code by a meta-model (see Section 2.5.4).

2.5.6. Stochastic finite elements

The stochastic finite element method (SFEM) is an important tool for modeling and propagating uncertainties in computational mechanics. In essence, the SFEM extends the FE method to account for uncertainties associated with the parameters of the modeled structure. A variety of methods accomplishing this task have been proposed, extended and reviewed previously (see e.g. [37,47,48].

The computational efficiency of SFEM analysis has been recognized as a critical aspect already for quite some time. Besides depending on the computational costs for performing the numerical computations required by the various SFEM algorithms, the computational efficiency is also greatly affected by more implementation-related factors. This is mainly because SFEM implementations by definition need to be integrated with external programs or with program modules performing FE analysis.

Since it is beyond the scope of this manuscript to provide the theoretical background of the various SFEM algorithms, it is referred to e.g. [38,48] for the formulations of the considered and well known methods within this context, such as the Neumann Expansion method [52], the Perturbation method [16] and Spectral SFEM, which utilizes the Polynomial Chaos (P-C) Expansion formulation [11]. P-C has been applied with success in several fields of engineering, see e.g. [4,20]. Instead, some of the key aspects considered within the development of this toolbox are to be explained briefly in the following part. The main capabilities of the SFEM toolbox are summarized in Table 1. As can be seen from the table, this toolbox is specifically designed to handle structural analysis problems only.

Table 1
Overview of the capabilities of the SFEM toolbox.

Solvers	NASTRAN, ABAQUS, ANSYS
Random parameters	Young's Modulus, density, shell elements thickness, beam elements cross sectional dimensions, force
Formulations	Perturbation, Neumann expansion, polynomial-chaos expansion
Implementations	Component-wise, solver-based, reduced model
Analysis types	Linear static, modal

For the detailed descriptions of the introduced implementations and algorithms, the reader is again referred to the references mentioned above.

Data management: The first step through the development of this toolbox has been to identify the bottlenecks within the SFEM analysis. These efforts have revealed that the data transfer between the 3rd party software (i.e. deterministic FE solvers) and the main driver program (i.e. SFEM Toolbox) plays a crucial role in this regard. The computational costs for this task become especially cumbersome, as the size of the FE model and the number of random parameters increase. In order to overcome this difficulty, optimized data handling concepts are developed within the toolbox. More specifically, one implementation exploits the fact that in engineering structures distinct components frequently have distinct properties (therefore called *component-wise* implementation); the second implementation concept greatly reduces the cost of data transfer by performing the linear algebra operations within the FE solver's programming environment (therefore called *solver-based* implementation, please refer to [26] for further details).

It should be noted that regarding the transfer of system matrices/vectors, various factors come into the play. In other words, different FE solvers export these quantities in different formats, e.g. *op2* or *op4* format (NASTRAN), *Harwell-Boeing* format (ANSYS) or *mtx* format (ABAQUS), etc. Clearly, one requires dedicated scripts to read and transfer these data into a meaningful format for the main program (in this case, into MATLAB matrices/vectors). Hence, the overall efficiency of this step becomes a function of all these parameters, namely the capability of the FE solver in exporting system quantities, efficiency of the utilized methods to import, format of the data, etc.

Parameter settings: As mentioned above, the user-friendliness is one of the key aspects of the toolbox. In this regard, it is aimed that also users without much experience in SFEM analysis can perform stochastic structural analysis. For this purpose, algorithms, which adjust the required input parameters automatically for a given problem, are developed. For example, the utilization of iterative solvers (e.g. Preconditioned Conjugate Gradient solver) and preconditioners (e.g. incomplete Cholesky factorizations) is necessary in order to handle the large size system of equations resulting from Galerkin based P-C Expansion formulations. Through the development of this toolbox, it has been observed that the input parameters for these tools affect the overall efficiency significantly. Therefore, novel strategies have been introduced, which determine adequate values for their input parameters, e.g. drop tolerance or convergence threshold (see [27] for details). It should be however noted that the proposed approaches do not remove the already existing drawbacks or limitations of the selected preconditioner, iterative solver or the P-C method itself. Instead, these algorithms aim to improve the user-comfort of the use of these tools, i.e. which enable to use these tools in a *black-box* fashion.

Since the efficiency of the analysis has been at the focus during the development, various strategies have been considered in order to reduce the computational costs to a minimum. One approach followed for this purpose has been to pre-calculate and store some quantities in a database. Instead of calculating these quantities at every analysis, they are then simply retrieved from the database. Examples to such quantities are the coefficients required to construct the system of equations arising from the P-C formulation or DMAP (Direct Matrix Abstraction Programming) scripts for NASTRAN, which are used to export the system matrices/vectors.

Deterministic model reduction techniques: The integration of deterministic model reduction techniques within the SFEM analysis has been considered in the toolbox. The proposed implementation is based on the idea that, if the quantities of interest to be investigated are limited to only certain critical components of the overall structure, the stochastic problem can be reformulated only for those locations. In other words, the system matrices can be reduced to the DOFs of interest only. This approach has been implemented for the P-C expansion formulation, where the size of the FE model is reduced using the Guyan reduction formulation. It is shown that due to the reduction in the size of the system of equations to be solved, a significant improvement in the overall efficiency can be achieved. This has also given rise to the application of the P-C method on structures of engineering practice, which might contain a substantial number of random parameters (see e.g. [28]).

2.5.7. High performance computing

Since all of the stochastic analysis methods require to perform an analysis multiple times, it is possible to decrease the overall analysis time by exploiting the inherent parallelism showed by the stochastic analysis algorithms on parallel computers.

In fact, computer industry has been rapidly transitioning from CPUs with a single processing core to multi-core configurations. Furthermore, in a matter of just a few years, the general-purpose computation on the Graphical Process Units (GPU) systems are already outperforming CPU-only clusters in many fields. The multi-core CPUs and the GPUs can be linked by forming a heterogeneous computing power network, i.e. a *grid computing*. High performance computing is nowadays largely available. Hence, efficient algorithms must take advantage of the availability of large number of cores and the heterogeneity of the grid computing, i.e. submitting a specific task to a specific hardware in order to reduce the wall-clock time significantly.

The described software offers the possibility to perform stochastic analysis adopting the immense computational power and the great opportunity provided by grid computing (e.g. [29,30]). In fact, grid computing offers at a relatively low costs more flexibility than the traditional parallel execution of the code in very expensive supercomputers [21]. Furthermore, by interfacing with grid managers (e.g. Oracle Grid Engine [49]) it is possible to distribute the execution of the analysis on the available (remote) resources on a computer grid and to maximize the use of the available licenses, while reducing the execution time (wall clock time) of the analysis task (see Fig. 5).

2.6. Competitive edges

The competitive edges of the general purpose software for stochastic analysis are briefly summarized in this section. The core components are based on Matlab environment, known for its

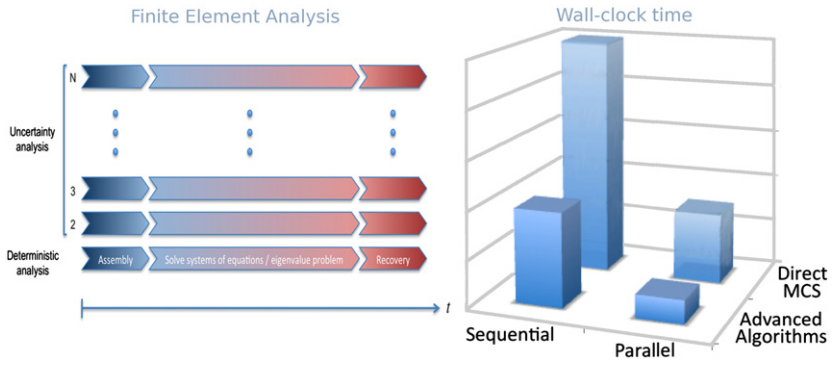

Fig. 5. Deterministic analysis vs uncertainty analysis and reduction of the computational wall-clock time adopting advanced algorithms and parallel computing.

Table 2
Main algorithms and procedures implemented in the general purpose software.

Toolbox	Main algorithms and procedures
Uncertainty	RVs, stochastic processes, random fields
Reliability	Advanced MC and approximate methods (FORM, bounds)
Sensitivity	Local and global sensitivity (Sobol' measures), bounds
Optimization	Genetic algorithms, COBYLA, simulated annealing, SQP
Meta-modeling	ANN, response surface, polynomial-chaos exp.
SFEM	Neumann/polynomial-chaos exp., perturbation method
HPC	Parallel/distributed computing licence optimization

highly optimized matrix and vector calculations. These components include several predefined solution sequences to solve a number of different problems. Typical applications include UQ and management, reliability based optimization and robust optimization, lifecycle management, model validation and verification.

In order to address these problems efficiently, the software includes the state-of-the-art of the algorithms and numerical procedures for simulation, reliability and optimization analysis, respectively. Table 2 shows the main algorithms and procedures implemented in the general purpose software considered here.

3. Numerical examples: large-scale FE models

In this section, the applicability of the proposed approaches to analyse problems of practical industrial interest is demonstrated. In Section 3.1, the capabilities of the SFEM toolbox are demonstrated analyzing a 5-storey building. In Section 3.2, the reliability analysis of a multi-storey building modeled with the 3rd-party software ABAQUS, ANSYS and NASTRAN is presented. In Section 3.3, an optimization analysis of a cylindrical shell modeled with the software ABAQUS is presented. Finally, in Section 3.4 the global sensitivity analysis of the GOCE satellite [10] modeled with the software NASTRAN is shown.

3.1. Uncertainty management of a multi-storey building

3.1.1. Description of the model

The following example is presented to describe how the effect of uncertainties within structural parameters of a large and complex FE model can be analyzed using the SFEM framework. More specifically, it is aimed to show that by utilizing advanced intrusive SFEM implementations, where the data management is optimized, the uncertainty management can be carried out at affordable computational costs.

In this numerical example a multi-storey building concrete structure (see Fig. 6), modeled with PATRAN, is selected as the deterministic model. The FE model contains 64,185 DOFs, where the floors and C-section walls in each floor have been modeled with shell elements. For modeling the girders and the columns, beam elements with rectangular cross sections have been used.

The building is fully constrained at the ground nodes and is loaded in $+y$ direction as shown in Fig. 6(b). In this example, the quantities of interest are selected as the y displacements at five different locations, i.e. Nodes 6710, 6485, 6260, 6035 and 9, respectively as shown in Fig. 6(c). These displacements reveal the inter-storey drift occurring between various floors, due to the loading and asymmetry of the structure. The loading has been assumed deterministic in order to investigate the effects of the uncertainties in the material properties solely. It should be noted that the SFEM analysis has been performed while choosing various different FE solvers, i.e. NASTRAN, ABAQUS and ANSYS, in order to present the capabilities of the toolbox in terms of interacting with various 3rd party solvers.

Regarding the stochastic part of the analysis, the effect of the uncertainties in the Young's modulus has been analyzed. These uncertainties are modeled by (truncated, i.e. negative, non-physical values ignored) normally distributed RVs, where each RV corresponds to the material of a certain component of the building. Fifteen RVs are considered in this example and the detailed description of this probabilistic model can be found in Table 3. The mean values of the Young's modulus (25 GPa) is a typical value for the grade 30 concrete [54].

3.1.2. SFEM analysis

The effect of the uncertain structural parameters on the response is investigated using the offered algorithms within SFEM toolbox, namely the Perturbation, Neumann Expansion and P-C Expansion formulations. It should be noted here that for the standard implementations, full size system matrices, i.e. 64,185 $\times$ 64,185, are transferred between the general purpose software and the FE solvers. Considering the *component-wise* implementation on the other hand, the different random parts of the building are considered separately. As a result, only the corresponding random components of the system matrices are transferred between software. Regarding the *solver-based* implementation, the transfer of the system matrices are avoided totally and instead only the quantities required for the post-processing are carried to the toolbox. Finally, in the case of the *reduced model* implementation (Guyan P-C), only the DOFs for which the variation of the displacements are sought, are processed in the analysis. As a result, the size of the deterministic model is reduced to 5 DOFs in this case.

3.1.3. Results and comments

To start with, the second-order statistics of the response estimated using the available algorithms are compared in order to verify the implementations within the toolbox. A fairly good

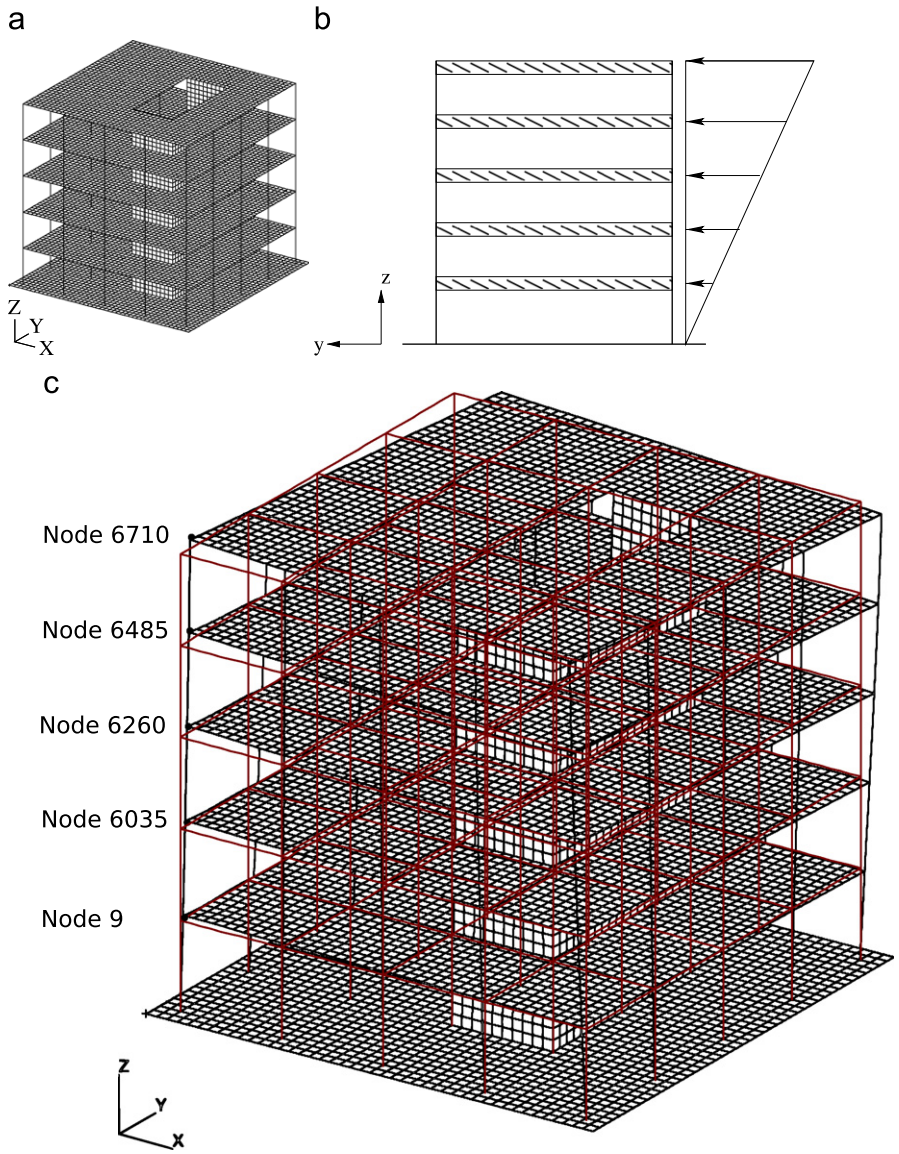

Fig. 6. (a) FE model of the multi-storey building model; (b) applied load and (c) deflection of the building due to the loading.

Table 3
Description of the probabilistic input model.

Random parameter	Mean value (GPa)	CoV
Young's modulus—columns (1 *RV per floor*)	25	0.15
Young's modulus—beams (1 *RV per floor*)	25	0.15
Young's modulus—C-section walls (1 *RV per floor*)	25	0.15

agreement between the implemented formulations and the direct MC simulation (reference results) as shown in Table 4. It should be also noted that the resulting CoV of the displacements are significantly smaller than the CoV of the input parameters. This indicates that the influence of the variability of the Young's modulus on the responses considered in this example is not very significant.

It should be noted that the SFEM toolbox allows also to compute the pdf of the quantity of interest from the samples of the Neumann expansion method or generating samples from the analytical expression obtained by the P-C expansion.

Computational aspects: As far as the efficiency of the SFEM analysis is concerned, the computational costs of the SFEM analysis

Table 4
Second order statistics of the response of the multi-storey building model.

Node ID	Perturbation (second order)	P-C (second order)	Neumann (100 samples)	DMCS (500 samples)
9	$\mu=0.0032$ CoV=0.070	$\mu=0.0032$ CoV=0.073	$\mu=0.0032$ CoV=0.076	$\mu=0.0032$ CoV=0.076
6035	$\mu=0.0054$ CoV=0.057	$\mu=0.0055$ CoV=0.060	$\mu=0.0055$ CoV=0.066	$\mu=0.0055$ CoV=0.060
6260	$\mu=0.0076$ CoV=0.051	$\mu=0.0077$ CoV=0.053	$\mu=0.0078$ CoV=0.054	$\mu=0.0077$ CoV=0.054
6485	$\mu=0.0095$ CoV=0.047	$\mu=0.0096$ CoV=0.049	$\mu=0.0097$ CoV=0.047	$\mu=0.0097$ CoV=0.049
6710	$\mu=0.0107$ CoV=0.044	$\mu=0.0109$ CoV=0.047	$\mu=0.0110$ CoV=0.045	$\mu=0.0109$ CoV=0.048

are summarized in Fig. 7. These have been reported in terms of CPU units, where each CPU unit corresponds to the time required for a single deterministic FE analysis (in this example, 1 CPU unit $\simeq 5$ s).

More specifically, Fig. 7(a) provides a comparison for the computational costs of the various methods and implementations

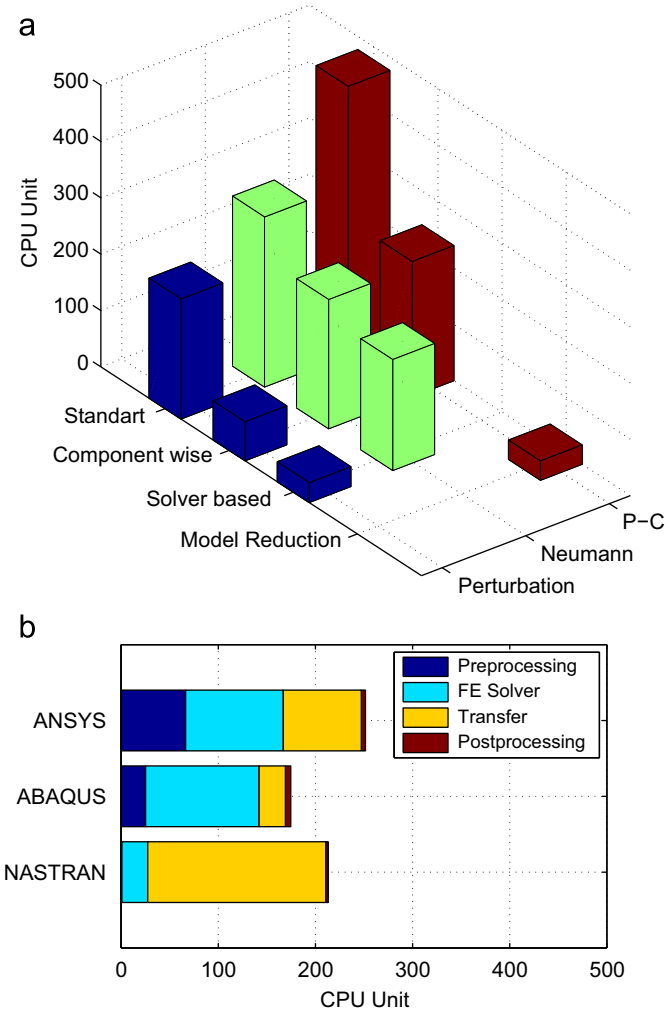

Fig. 7. (a) Computational costs of the SFEM analysis considering various formulations and implementations; (b) computational costs for the standard perturbation implementation considering various commercial FE solvers.

offered by the SFEM toolbox. It should be noted that the results presented in this plot have been obtained, while NASTRAN has been used as the deterministic FE solver. One can note here the substantial computational savings thanks to the optimized implementations. Considering for example the component-wise implementations, the overall analysis times are much less than the standard implementations, since the amount of data transfer is significantly less. The same argument applies also to the solver-based and reduced model implementations. However, it should be pointed out that for the reduced model implementation, in addition to the reduced data transfer, also the size of the system of equations is significantly smaller.

Fig. 7(b) shows the computational costs for the standard Perturbation implementation while selecting different FE solvers as the 3rd party software. It should be remarked at this point that these results are not meant to provide a measure for the efficiency of these solvers. Instead as mentioned previously, the motivation here is to show the interaction capability of the SFEM toolbox with different solvers. Furthermore, this plot is meant to provide information about the different stages of the SFEM analysis. For examples, the *preprocessing* stands for the part, where the various deterministic input files are prepared for the solvers according to the probabilistic model. Then, the FE solver is executed for each of these input files (*FE Solver* stage) in order to export the required system matrices/vectors. The third stage corresponds to the *transfer* part, where these quantities are transferred to the toolbox. Finally, in the *postprocessing* part, these transferred system quantities are processed according to the selected formulation. As a result, the second-order statistics are estimated for the structural responses. In summary, the CPU times reported here are affected by many factors, such as the time required to execute the solver, the preparation of the input files, the efficiency of the tools used to transfer system matrices from different formats, etc.

3.2. Reliability analysis of a multi-storey building

In the second example, a multi-storey building modeled with ABAQUS (shown in Fig. 8) is used for the application of efficient reliability analysis. The load case under investigation is the combination of (simplified) lateral wind load and self weight. The former load type is modeled by deterministic concentrated static forces acting on the nodes of one edge of each floor and on the upper part of the staircase (indicated by the small arrows in Fig. 8), where the magnitudes increase with the height of the building. Failure is defined as the exceedance of the yield stress in a bar element of one column of the fifth floor marked in red in Fig. 8, where the yield curve is defined by the Tresca criterion. Hence, the performance function is defined by

$$g(\theta) = \sigma_{\max} - \sigma(\theta) \tag{11}$$

where σ_{max} defines the maximum stress level and $\sigma(\theta)$ is the element stress as extracted from the output file of the FE-analysis using the structural parameter vector θ.

The FE-model of the structure involves approximately 8200 elements and 66,300 DOFs, where solid elements (C3D8I) are used for the foundation, the mesh of the floors consists solely of quadrilateral elements (S4) and each of the 16 columns of all floors are modeled with 2-node beam elements (B31). In order to consider the uncertainties within the structural parameters, a total number of 244 independent RVs are used. More precisely, a Gaussian distribution is used to represent the resistance, i.e. the maximum allowable stress, while log-normal distributed RVs are assigned to the Young's modulus, the density and the Poisson ratio of the columns, the walls and the stairs. In addition, the cross sectional properties of the columns are modeled by uniform distributions assigned to both the width and the height. A summary of the distribution parameters characterizing the stochastic analysis is listed in Table 5. The shear moduli are defined as functions of the RVs representing the respective Young's moduli and Poisson ratios.

3.2.1. Sensitivity and reliability analysis

The solution strategy adopted to estimate the failure probability of the multi-storey building is depicted in Fig. 9. This

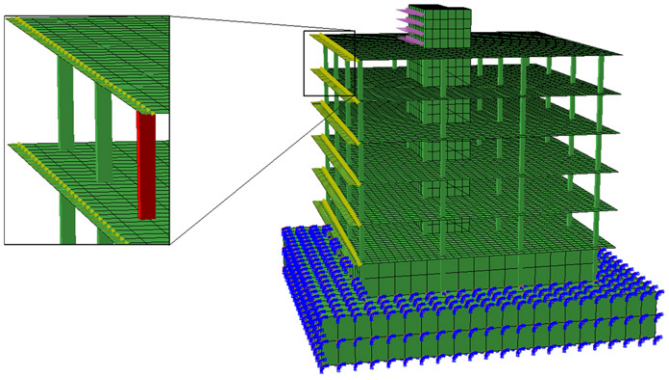

Fig. 8. FE-model of a multi-storey building with indication of the loading and the observed column of interest (showed in red color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Random variables used for modeling the uncertainties within the multi-storey building.

RV IDs	Distribution	Component property
1	Gaussian (1.0E−8, 1.0E−7) Pa	Column resistance
2-193	Uniform [0.36, 0.44] m	Columns height and width section
194-212	Lognormal (3.5E−10, 3.5E−9) Pa	E-modulus of columns, floors, stairs
213-231	Lognormal (2500, 250) kg/m^3	Density of columns, floors, stairs
232-244	Lognormal (0.25, 0.025)	Poisson ratio of floors, stairs

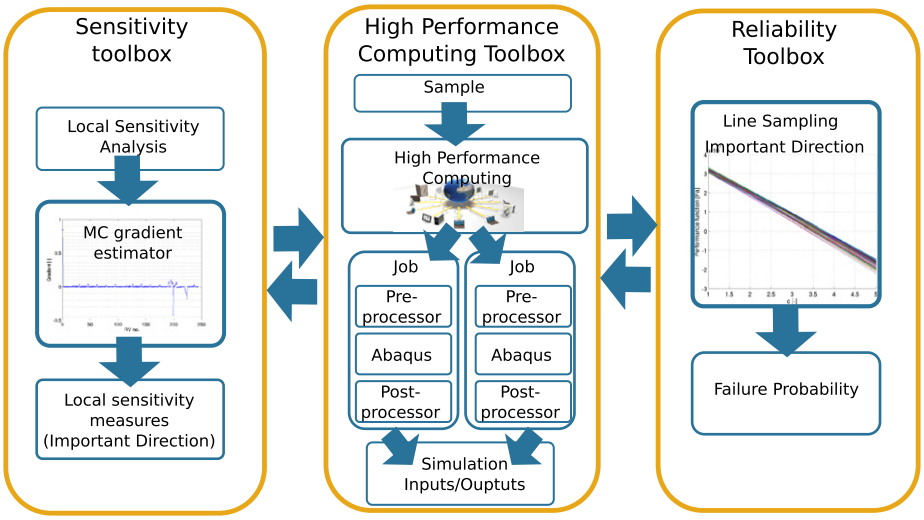

Fig. 9. Solution strategies adopted to estimate the failure probability of the multi-storey building.

estimation using direct MC simulation of this large FE-model may become unfeasible due to the large sample size needed to trustworthy identify the failure region. Hence, an advanced sampling method, namely Line Sampling [17,43], is adopted here. The procedure requires a priori information on the important direction pointing towards the failure domain. Therefore, as a first step, the gradient at the origin of the standard normal space is determined by means of finite differences (Sensitivity toolbox). Then, the Line Sampling method (Reliability toolbox) is used by passing the computed gradient.

Since the samples needed for the finite differences are not dependent on each other, the determination of the gradient is parallelized by employing the features of the High Performance toolbox. In addition, also the parallelism associated with the possibility to concurrently evaluate the responses along the lines when performing Line Sampling is exploited.

In the following, the results of the parallelized reliability analysis are discussed.

3.2.2. *Results and comments*

The importance of the random structural parameters with respect to the performance function is evaluated by investigating the amplitude of the gradient in the 244-dimensional standard normal space. The components of the unit vector of the gradient $\boldsymbol{\alpha}$ are shown in Fig. 10(a), where the first component is the resistance, the components between 2-193 concern the cross sectional properties, the components 194-212 the Young's moduli, the components 213-231 the density and the remaining components 232-244 the Poisson ratios (see also Table 5). From this figure, it can be observed that the performance function is governed by the resistance and also the Young's modulus of the columns in floor 6 (component no. 199). Furthermore, the densities of the columns in floor 5 and 6 (components no. 224 and 223) reveal a dominant influence.

The results of Line Sampling are shown in Fig. 10(b), where the values of the performance function as defined in Eq. (11) are plotted. Each curve corresponds to the responses evaluated along one of the lines in the direction of $\boldsymbol{\alpha}$. The responses have been evaluated at $c\boldsymbol{\alpha}$, where for each line, c takes the values of $(1, 2, \ldots, 5)$. Between these points, the responses have been approximated by using splines. The pronounced downwards trend of all lines confirms that the important direction has been identified. In addition, the sensitivity of the performance function with respect to variations in the direction orthogonal to $\boldsymbol{\alpha}$ is small. Based on this observation, it can be concluded that the non-linearities of the performance function with respect to the RVs are small and therefore a high accuracy in the estimation of the failure probability can be achieved (in the ideal, i.e. linear case, all lines would overlap). The resulting failure probability of $p_f = 1.4 \times 10^{-4}$ is obtained from the mean value of the resulting values of c leading to the intersections with the performance function. The associated coefficient of variation of the estimated p_f results to $Cov = 0.044$.

Computational aspects: The computation of the gradient $\boldsymbol{\alpha}$ is amenable to parallel processing due to the independence in the evaluation of the single components. However, in practice, this parallelization is usually limited by the number of available licenses for the FE-solver. Here, in order to represent this, a total number of three ABAQUS licenses have been used in the present example. The thereby resulting time needed for the evaluation of the gradient amounts up to $t_{\text{par}} = 59$ min, which constitutes a significant reduction to the wall clock time needed in a sequential analysis ($t_{\text{seq}} = 148$ min). The achieved speed-up $S = t_{\text{seq}}/t_{\text{par}} = 2.5$ is however smaller than the ideal (linear) speed-up of 3.0. The main reasons can be found in the differences in the hardware architecture, in the queuing time and also in the sequentially executed extraction of the response from the FE-output files.

In case of direct MC simulation, a total number of $N_s = 3.7 \times 10^6$ samples would have been required in order to reach the same

coefficient of variation as obtained by the Line Sampling method, which would exceed a feasible analysis time. In the present case, the number of model evaluations needed for performing Line Sampling is $N_s = 500$ leading to the total analysis time of $t_{LS,seq} \approx 5$ h. If this advanced reliability method is further combined with parallel processing, the wall clock time decreases remarkably to $t_{LS,seq} \approx 1.9$ h, leading to a similar speed-up as for the determination of the gradient.

In conclusion, it can be seen that by utilizing advanced simulation methods together with the parallel computing features, reliability assessment of realistic models can be performed for reasonable computational costs.

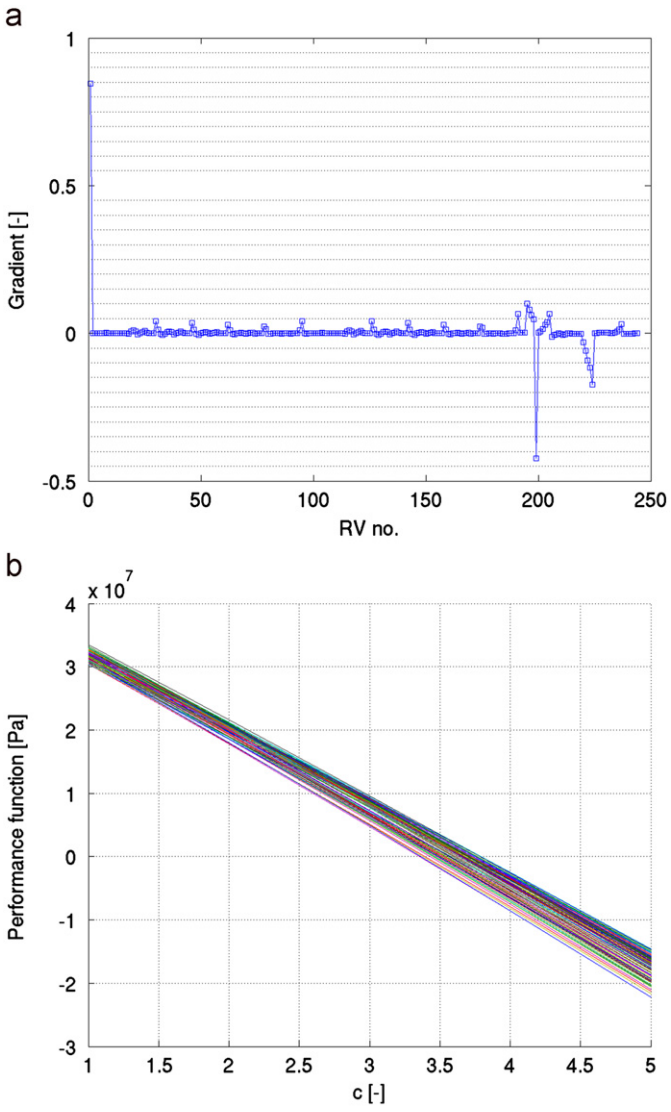

Fig. 10. Graphical representation of the (a) 244 components of the gradient unit vector and (b) line sampling of the performance function, respectively.

3.3. Robust optimization of a cylindrical shell

The third numerical example aims to perform robust design of a cylindrical shell under deterministic axial compression. The aim of the design optimization is to minimize the total weight of the cylinder, while choosing cylinder radius and shell thickness as design parameters. Geometric imperfections of the cylinder surface are also considered in the analysis, i.e. the geometry of the structure slightly differs from the shape of a perfect cylinder. Thus, the buckling load shows a high variability even for cylinder of the same design, with buckling loads lower than the one predicted by classic analysis. In order to take into account the variability of the buckling load for each proposed design, a constraint derived from the so-called *Design-for-Six-Sigma* [53] is introduced for the normalized buckling load, λ. The normalized buckling load is the ratio of the buckling load of the imperfect cylinder and of the buckling load given by the classic analytic formula. The constraint of the optimization is thus imposed as

$$\mu(\lambda) - 4\sigma(\lambda) \geq \lambda_{lim} \tag{12}$$

where $\mu(\lambda)$ and $\sigma(\lambda)$ are the mean and the standard deviation of the buckling load respectively. The selected load threshold level for this analysis is $\lambda_{lim} = 0.1$. Assuming that the buckling load is normally distributed, the limit in Eq. (12) corresponds to a probability of 3×10^{-5}.

A visualization of the applied solution strategy is presented in Fig. 11.

3.3.1. Description of the problem

A FE model of the cylindrical shell have been constructed to compute the buckling load. The cylinder shows isotropic material properties with a Young modulus equal to 104.4 GPa and a Poisson ratio equal to 0.3. The geometry is meshed with membrane, finite strain quadrilateral shell elements and has in total 49,000 degrees of freedom. A vertical displacement is imposed at the extremities of the cylinder and the reaction forces at an

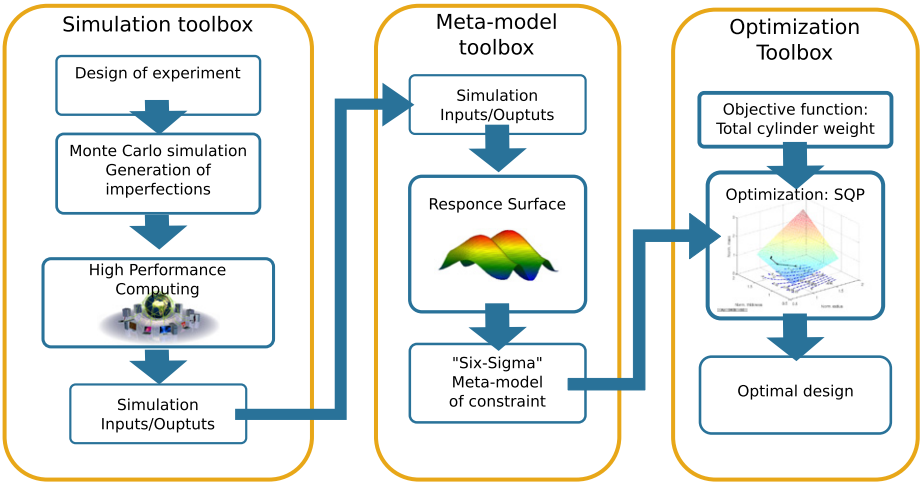

Fig. 11. Solution strategies adopted to optimize the cylindrical shell.

extremity of the cylinder are computed, obtaining a force–displacement relation. The buckling load is defined as the first maximum of the force–displacement function. Once buckling is initiated, the reaction force drops. FE simulation is performed using a stabilized Newton–Raphson scheme in ABAQUS considering geometrical non-linearity.

A database of imperfections of seven aluminium cylindrical shell has been obtained from a study of TU-Delft [2]. These experimental data have been fitted to a regular grid, common to all cylinders, in order to describe the surface imperfections with a two-dimensional random field. This random field represents the variation of the surface of the cylinder from the nominal radius as a function of the circumferential coordinate and of the axial coordinate. In this case, the covariance matrix is directly computed from the experimental data with no additional assumption, obtaining a non-homogeneous Gaussian random field. New samples of the random field are obtained with the Karhunen–Loève expansions [14,19]. Since the covariance matrix is established directly on the measured imperfections only very few eigenvalues are larger than zero. The imperfections are inserted into the input file as deviation of the nodal coordinates of the perfect FE model nodes.

For each proposed cylinder design, the mean and standard deviation of the buckling load of imperfect cylinders have been computed by means of MC simulation. Each MC simulations is performed using 100 samples where the imperfections show identical statistical properties.

3.3.2. Meta-modeling and optimization

In this example, the value of the radius and of the thickness have been normalized to their nominal values, which are respectively 1.016×10^{-2} and 1.016×10^{-4} m. In order to reduce the computational time of the optimization, a response surface of the constraint function has been trained, considering the thickness and the radius of the cylinder as inputs. This meta-model approximates the non-linear constraint of the buckling load. A full quadratic response surface has been trained using nine support points, each representing nine different candidate designs. These designs are shown in Table 6. Furthermore, the lower bounds (0.8) and upper bounds (1.5) of the normalized radius of the cylinder and of the normalized thickness are considered as additional constraints, respectively.

The optimization is performed using *Sequential Quadratic Programming*. The efficiency of the proposed approach has been verified by repeating the optimization procedure starting from various different initial points.

3.3.3. Results and comments

The cylinder designs used as calibration points of the response surface have been characterized totally by their buckling behavior. Table 6 shows the mean and the standard deviation of the buckling load w.r.t. design parameters. The radius of the cylinder has more influence on the buckling load than its thickness. The response surface of the non-linear constraint is shown in Fig. 12(a). Fig. 12(b) shows the search of the optimum done by the sequential quadratic programming algorithm, while taking the nominal cylinder as initial design. The optimum has been found at a normalized radius of 0.826 and a normalized thickness of 1.46. For this configuration, the normalized weight is approximately 1.2 and the non-linear constraint on the buckling load is satisfied.

The optimization process was repeated several times with initial design parameters taken randomly within the area of interest. All the optimization schemes converged to the same optimum. Fig. 13 shows the evolution of the objective function and non-linear constraint during optimization.

The use of response surface allows performing the optimization efficiently within affordable computational time. Indeed, the gradient of the buckling load needs to be computed accurately. Each MC simulation lasts approximately 50 h on a dual quad-core Intel Xeon E5430 machine with 8 Gbytes of RAM. Two concurrent executions of the ABAQUS solver were performed. Each FE computation was additionally parallelized using four CPUs. The overall computational time required to train the response

Table 6
Normalized mean and standard deviation of the buckling load obtained at the design of experiment points.

Normalized radius and thickness	Normalized buckling load
(1, 1)	$\mu = 0.3851$, $\sigma = 0.0791$
(0.8, 0.8)	$\mu = 0.3606$, $\sigma = 0.0835$
(0.8, 1.5)	$\mu = 0.4097$, $\sigma = 0.0783$
(1.5, 0.8)	$\mu = 0.4295$, $\sigma = 0.1113$
(1.5, 1.5)	$\mu = 0.4412$, $\sigma = 0.0742$
(0.717, 1)	$\mu = 0.3793$, $\sigma = 0.0831$
(1, 0.717)	$\mu = 0.3703$, $\sigma = 0.0897$
(1.707, 1)	$\mu = 0.4031$, $\sigma = 0.0713$
(1, 1.707)	$\mu = 0.4210$, $\sigma = 0.0847$

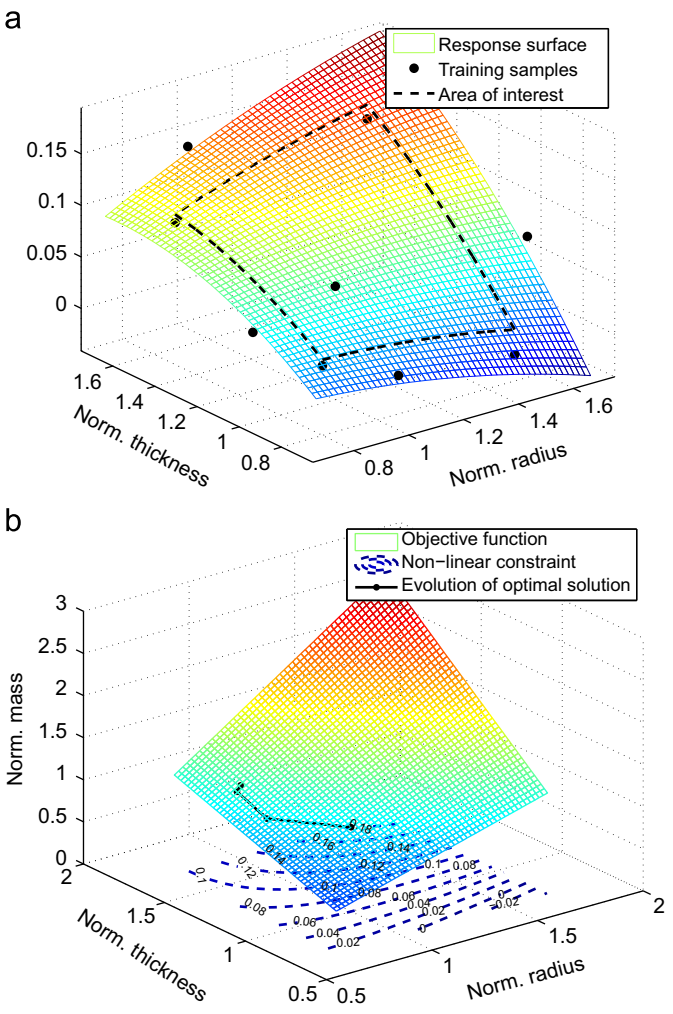

Fig. 12. (a) Response surface of the buckling load with respect to the radius and thickness of the cylinder. The dash black line shows the upper and lower bounds which are considered for the thickness and radius of the cylinder. (b) Evolution of the optimal mass during the optimization procedure. The dashed (blue) lines at the bottom are the contour levels of the non-linear inequality constraint. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

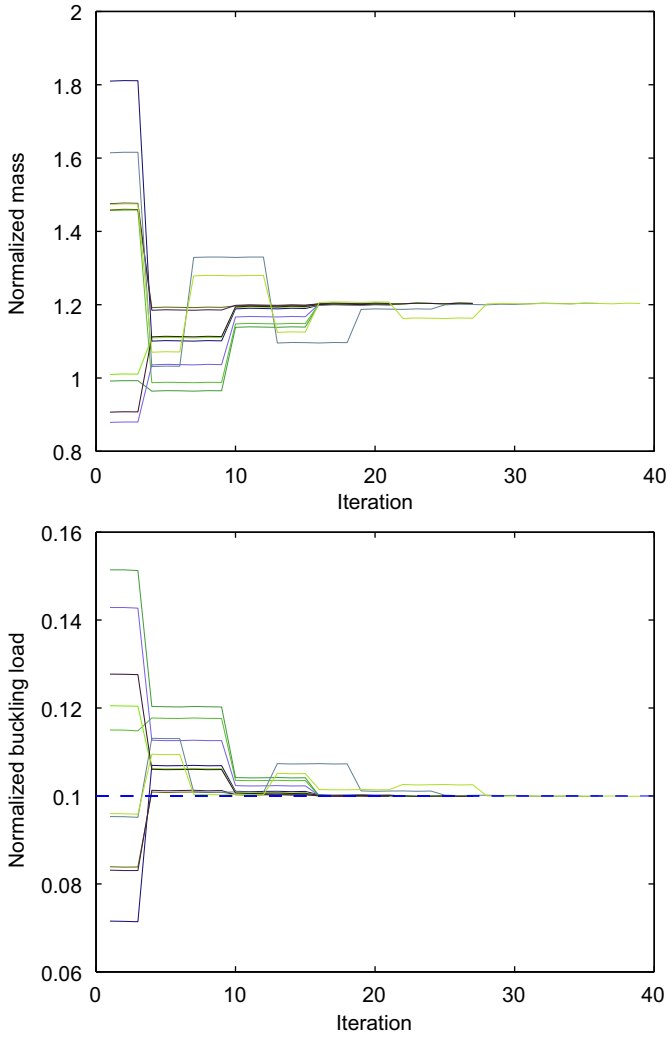

Fig. 13. Evolution of the properties of the cylinder during the optimization for various starting points. (a) Normalized weight of the cylinder with respect to the optimization iterations (evaluations of the objective function). (b) $\mu(\lambda)+4\sigma(\lambda)$ with respect to the optimization iterations (evaluations of the constraint function).

surface is equal to 18 days approximately, whereas approximately 60 days would have been necessary to perform one optimization using the full model instead of the meta-model. It is clear that if more licenses would have been available, the simulation time could have been further reduced.

3.4. GOCE satellite

3.4.1. Description of the problem

In the fourth numerical example, sensitivity analysis with respect to modal properties is applied to a full satellite model as shown in Fig. 14. It involves the Gravity Field and Steady-State Ocean Circulation Explorer (GOCE) satellite, whose aim is the determination of the geoid and to measure the gravitational field of Earth with a very high degree of accuracy in a low Earth orbit.

The structure is modeled using MSC.Patran/Nastran where the FE-model has been provided by Thales Alenia Space Italy. The model involves a total number of approximately 360,000 DOFs ($\approx 74,000$ elements) and consists of a main satellite platform and a gravitational gradiometer. In the main GOCE platform, quadrilateral (QUAD4) and triangular (TRIA3) shell elements are used to model the body panels, the wings, the winglets, the internal

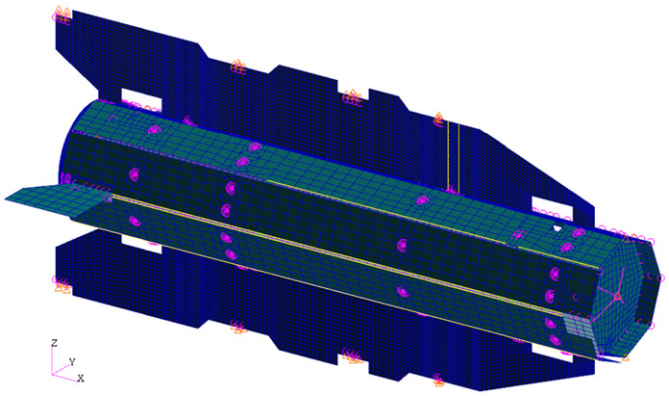

Fig. 14. FE model of the GOCE satellite (courtesy of Thales Alenia Space Italy).

Table 7
Definition of the groups of parameters in the GOCE satellite.

Group no.	Parameters
1	Young's modulus of isotropic materials
2	Poisson's ratio of isotropic materials
3	Young's modulus in the principal direction of orthotropic materials
4	Young's modulus in the secondary direction of orthotropic materials
5	Poisson's ratio of orthotropic materials
6	In-plane shear modulus of orthotropic materials
7	First out-of-plane shear modulus of orthotropic materials
8	Second out-of-plane shear modulus of orthotropic materials
9	Densities of the materials
10	Thicknesses of the shells
11	Linear elastic connections of panels to the main satellite structure
12	Linear elastic connections of panels to the satellite wings
13–18	Linear elastic connections of the wings to the main satellite structure

floors and the solar panels. Beam elements (BAR, ROD and BEAM) constitute the connections of the wings to the main structure and of the instrumentation to the floors. Solid elements (HEXA and PENTA) are used in the Launch Vehicle Adapter (LVA) ring, and scalar spring elements (CELAS2) represent the connection between the solar panels and the structure, as well as the fixing of the wing to the main octagonal body.

A total number of 18 groups combining 3047 structural parameters are defined according to the type and location of the respective materials or geometric specifications as shown in Table 7. This grouping is carried out in order to reduce the number of parameters, since an independent processing of all involved structural parameters might not have been feasible for sensitivity analysis. Following this strategy, the most important parameter groups, and consequently the most important parameters can be determined.

3.4.2. Meta-modeling and sensitivity analysis

The global sensitivity analysis requires a substantial number of model evaluations. Therefore, it is not feasible to apply it directly for the GOCE model since each FE analysis requires approximately 3 min. In order to overcome this drawback the solution strategy shown in Fig. 16 has been adopted.

First an ANN has been calibrated and verified in order to replace the computationally expensive FE model and to approximate the unknown relationships between the modal properties of the satellite and the variables defined in Table 7. More

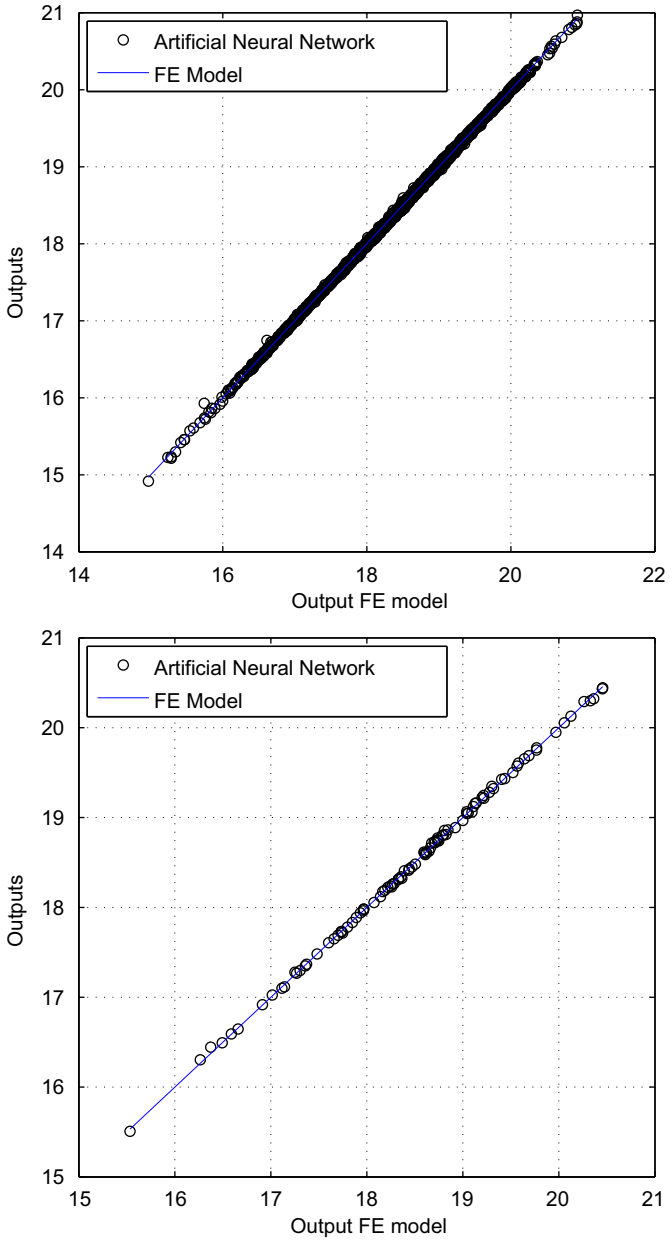

Fig. 15. Regression plot for the first eigenfrequency of (a) calibration data and (b) verification data, respectively.

specifically, a set of $N_{data} = 2000$ FE simulations have been performed with randomly perturbed structural parameters by means of MC simulation. Among these samples, 1900 have been dedicated to calibrate the ANN, while 100 have been used to verify the generalization capabilities of the trained ANN. Each ANN, constituted by $N_{inp} = 18$ inputs and $N_{out} = 1$ output, is dedicated to the prediction of either one of the first 24 eigenfrequencies or one of the diagonal values of the Modal Assurance Criteria (MAC) matrix.

An automated training procedure has been implemented such that various network topologies are tested. Only the best networks, characterized by the highest R^2 value (see Eq. (8)) on verification data, are kept. As an indication of the fairness of the network the regression plot for the ANN for the first eigenfrequency is shown in Fig. 15. The values of R^2 for the first seven frequencies and MAC values are listed in Table 8. It can be seen that all the ANN have very good generalization accuracy, except for the eigenfrequency #5 and diagonal MAC matrix term #7. However, the performance of these two ANN are still within acceptable range.

Having constructed a fast and sufficiently accurate meta-model by using ANN, global sensitivity analysis can be performed efficiently adopting the Latin Hypercube Sampling technique and the algorithms proposed in Ref. [36]. This efficient algorithm allows to estimate the main effect (first order index) and the total effect, respectively. A sample set of 6×10^6 samples have been used to estimate the sensitivity indices in approximately 2 h thanks to the established meta-model.

3.4.3. *Results and comments*

The estimated main and total effect for the first MAC values and the frequency of the first mode are shown in Figs. 17(a) and (b), respectively. These figures reveal that the first MAC values and eigenfrequency are mostly influenced by the thicknesses of the shells (group number 10) and the Young's modulus in the principal direction of orthotropic materials (group number 3) Fig. 18.

Computational aspects: The large number of analyses required by the global sensitivity analysis leads to high computational

Table 8
R^2 values of the first 14 ANN.

ANN output	R^2 of verification data
Frequencies 1–7	0.9996, 0.9997, 0.9997, 0.9996, 0.8830, 0.9780, 0.9582
Diag. MAC values 1–7	0.9989, 0.9978, 0.9958, 0.9986, 0.9495, 0.9762, 0.8278

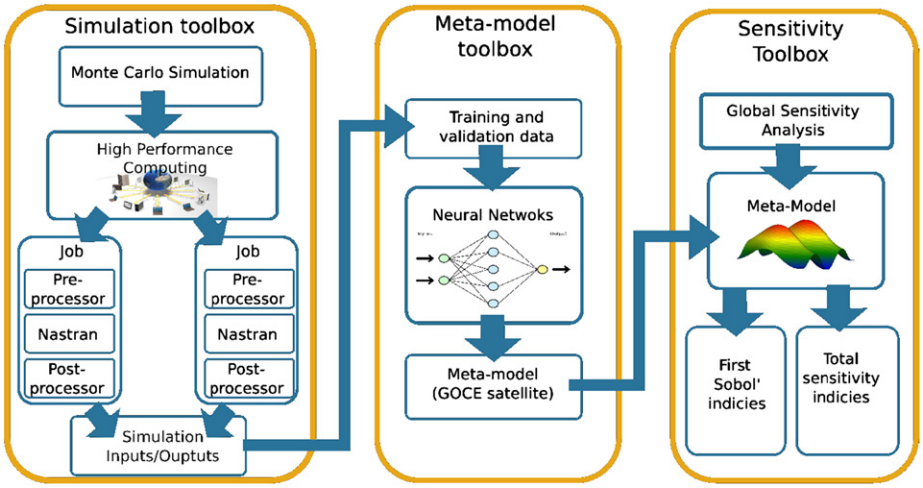

Fig. 16. Solution strategies adopted to perform the sensitivity analysis of the GOCE satellite.

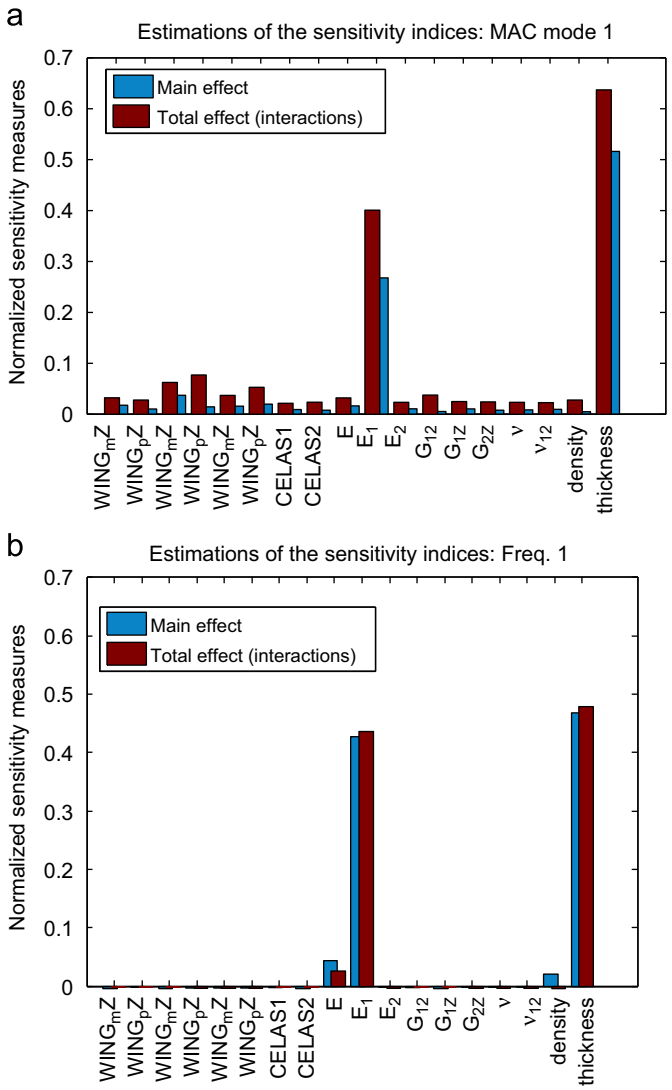

Fig. 17. Sensitivity measures analysis of the GOCE satellite obtained by means Sobol' approach for: (a) the MAC values and (b) the frequency of the first mode, respectively.

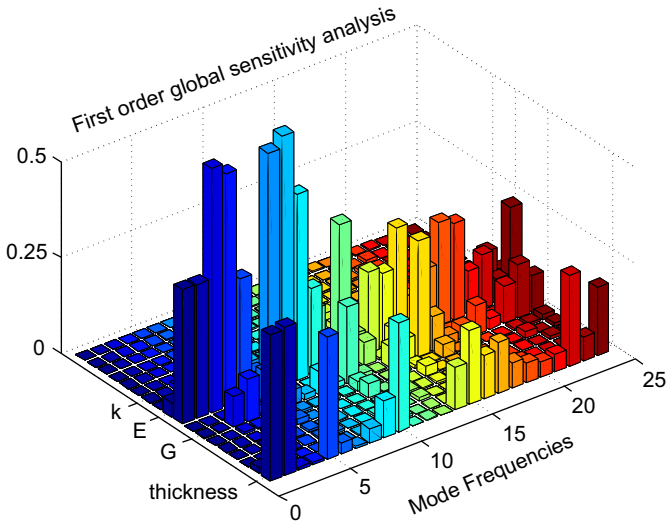

Fig. 18. First-order sensitivity analysis of the GOCE satellite obtained by means the random design variable approach for the first 24 modes.

efforts, making this kind of analysis infeasible for large and complex FE models.

On the employed computer facilities, namely on a dual quad-core Intel Xeon E5430 server with 8 Gbytes of RAM, the replacement of the nominal parameter values by random samples in the FE-input file, the normal mode analysis of the full model performed with MSC.Nastran and the import of the modal quantities into Matlab requires ≈ 220 s for one single normal mode analysis. Consequently, then the total analysis time needed for 6×10^6 model evaluations would have required a (theoretical) time of ≈ 41 years.

In the present example however, the expensive computational analysis is replaced by a surrogate model, which requires only 0.0014 s for each evaluation. As a result, by utilizing the advance meta-modeling techniques, global sensitivity analysis of a large a complex model has been made feasible (analysis time could be remarkably reduced to ≈ 2.3 h).

4. Conclusions

In this work it has been shown that stochastic analyses of large FE models of practical engineering interest are feasible. These kind of analyses are made feasible thanks to a general purpose software for stochastic analyses. The presented software allows to perform UQ, sensitivity analysis, optimization, reliability analysis and life cycle management study coupling the 3rd party FE solver with the state-of-the-art of the reliability and optimization algorithms. In this way the analyst can continue to use the deterministic FE-model that he/she is already familiar with. In fact, it appears difficult to otherwise adhere to the state-of-the-art, in terms of deterministic FE-modeling. This, in fact, is a key aspect to strengthen the bridge between the industry and academic researchers, since nowadays the popularity of any method used on real applications is very much dependent on the availability of efficient software, which facilitates the use of these methods without an extensive training.

The applicability of proposed general purpose software for handling large scale problems in terms of the number of uncertain parameters involved and the complexity of the FE models has been demonstrated by means of a number of different numerical examples.

In conclusion, the general purpose software provides the necessary tools to bring down the gap between the deterministic and stochastic analysis for practical applications. Combining the efficiency of advanced stochastic methods and the computational resources provided nowadays by high performance computing, the computational costs of a stochastic analysis can be significantly reduced. It is crucial that the stochastic tools and procedures are offered to the industry within an easy-to-use general purpose software, so that these algorithms are actually used for practical applications and its remedies can be recognized by the authorities.

Acknowledgments

The financial support of the Translational Research Project L269-N13 of the Austrian Science Foundation (FWF) is deeply appreciated by the authors. The fourth author is a recipient of a DOC-fForte fellowship of the Austrian academy of Science. We gratefully acknowledge M.F. Pellissetti, L. Pichler, M.A. Valdebenito, P. Furegato, G. Schulze and H. Marth for their contributions during the development of the software as described here.

References

[1] J. Anderson, Introduction to Neural Network, MIT Press, Cambridge, MA, 1995.

[2] J. Arbocz, H. Abramovich, The Initial Imperfection Data Bank at the Delft University of Technology, Part 1, Technical Report, Delft University of Technology (1979).
[3] J. Arora (Ed.), Optimization of Structural and Mechanical Systems, World Scientific, 2007.
[4] B.V. Asokan, N. Zabaras, A stochastic variational multiscale method for diffusion in heterogeneous random media, J. Comput. Phys. 218 (2) (2006) 654–667.
[5] S.-K. Au, J.L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probab. Eng. Mech. 14 (4) (2001) 263–277.
[6] I. Babuska, R. Tempone, G.E. Zouraris, Galerkin finite element approximations of stochastic elliptic partial differential equations, SIAM J. Numer. Anal. 42 (2) (2004) 800–825. doi:10.1137/S0036142902418680.
[7] F. Campolongo, S. Tarantola, A. Saltelli, Tackling quantitatively large dimensionality problems, Comput. Phys. Commun. 117 (1999) 75–85.
[8] W. Cauer, Theorie der linearen Wechselstromschaltungen, vol. 1, Akad. Verlags-Gesellschaft Becker und Erler, Leipzig, 1941.
[9] O. Ditlevsen, O. Madsen, Structural Reliability Methods, John Wiley and Sons, 1996.
[10] M. Drinkwater, R. Haagmans, D. Muzi, A. Popescu, R. Floberghagen, M. Kern, M. Fehringer, The GOCE gravity mission: Esa's first core earth explorer, in: Proceedings of Third International GOCE User Workshop, ESA SP-627, Frascati, Italy, 6–8 November 2006, 2007, pp. 1–8, ISBN 92-9092-938-3.
[11] R. Ghanem, P. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, Berlin, 1991.
[12] B. Goller, H.J. Pradlwarter, G.I. Schuëller, Robust modal updating with insufficient data, Comput. Methods Appl. Mech. Eng. 198 (37–40) (2009) 3096–3104. doi:10.1016/j.cma.2009.05.009.
[13] O. Hasancebi, Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures, Comput. Struct. 86 (1–3) (2008) 119–132.
[14] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, Am. Acad. Sci., Fennicade, Ser. A 37 (1947) 3–79.
[15] D. Kececioglu, Reliability Engineering Handbook, vol. 1, Prentice Hall, New Jersey, 1991.
[16] M. Kleiber, T. Hien, The Stochastic Finite Element Method: Basic Perturbation Technique and Computer Implementation, John Wiley & Sons Ltd., 1992.
[17] P.S. Koutsourelakis, H.J. Pradlwarter, G.I. Schuëller, Reliability of structures in high dimensions, part I: algorithms and applications, Probab. Eng. Mech. 19 (4) (2004) 409–417. doi:10.1016/j.probengmech.2004.05.001.
[18] P.-L. Liu, A. Der Kiureghian, Multivariate distribution models with prescribed marginals and covariances, Probab. Eng. Mech. 1 (2) (1986) 105–112.
[19] M. Loève, Probability Theory, 4th ed., Springer-Verlag, New York, 1977.
[20] D. Lucor, C.-H. Su, G.E. Karniadakis, Generalized polynomial chaos and random oscillators, Int. J. Numer. Methods Eng. 60 (2004) 571–596.
[21] F. Magoules, J. Pan, K. Tan, A. Kumar, Introduction to Grid Computing, Chapman & Hall/CRC Numerical Analysis and Scientific Computing, Taylor & Francis, 2009 ISBN-13: 9781420074062.
[22] R.E. Melchers, Structural Reliability: Analysis and Prediction, Wiley, 2002.
[23] R.H. Myers, D.C. Montgomery, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 2nd ed., John Wiley and Sons, Inc., New York, NY, 2002.
[24] J. Myung, Tutorial on maximum likelihood estimation, J. Math. Psychol. 47 (1) (2003) 90–100.
[25] B. Möller, M. Beer, Fuzzy-Randomness—Uncertainty in Civil Engineering and Computational Mechanics, Springer-Verlag, 2004.
[26] H.M. Panayirci, M.F. Pellissetti, Efficient component-wise and solver-based intrusive SFEM analysis of complex structures, Finite Elem. Anal. Des. 46 (2010) 449–459. doi:10.1016/j.finel.2010.01.009.
[27] H.M. Panayirci, Efficient solution for Galerkin based polynomial chaos expansion systems, Adv. Eng. Software 41 (412) (2010) 1277–1286. doi:10.1016/j.advengsoft.2010.09.004.
[28] H. Panayirci, H. Pradlwarter, G. Schuëller, Efficient stochastic finite element analysis using Guyan reduction, Adv. Eng. Software 42 (4) (2011) 187–196.
[29] M. Pellissetti, Parallel processing in structural reliability, J. Struct. Eng. Mech. 32 (1) (2009) 95–126.
[30] M.F. Pellissetti, G.I. Schuëller, Scalable uncertainty and reliability analysis by integration of advanced Monte Carlo simulation and generic finite element solvers, Comput. Struct. 87 (13–14) (2009) 930–947. doi:10.1016/j.compstruc.2009.04.003.
[31] E. Patelli, H.J. Pradlwarter, G.I. Schuëller, Global sensitivity of structural variability by random sampling, Comput. Phys. Commun. 181 (2010) 2072–2081. doi:10.1016/j.cpc.2010.08.007.
[32] E. Patelli, H.J. Pradlwarter, G.I. Schuëller, On multinormal integrals by importance sampling for parallel system reliability, Struct. Safety 33 (1) (2011) 1–7. doi:10.1016/j.strusafe.2010.04.002.
[33] L. Pichler, G.I. Schuëller, On the treatment of finite element structures in stochastic linear dynamics using a mode-based meta-model, Eng. Struct. 33 (2) (2010) 391–400.
[34] H.J. Pradlwarter, Relative importance of uncertain structural parameters, part I: algorithm, Comput. Mech. 40 (4) (2007) 627–635. doi:10.1007/s00466-006-0127-9.
[35] A. Saltelli, K. Chan, E. Scott, Sensitivity Analysis, Wiley Series in Probability and Statistics, vol. 535, John Wiley and Sons, Chichester, 2000.
[36] A. Saltelli, Making best use of model valuations to compute sensitivity indices, Comput. Phys. Commun. 145 (2002) 280–297.
[37] C.A. Schenk, G.I. Schuëller, Uncertainty Assessment of Large Finite Element Systems, Lecture Notes in Applied and Computational Mechanics, vol. 24, Springer-Verlag, Berlin/Heidelberg/New York, 2005 ISBN: 978-3-540-25343-3.
[38] G.I. Schuëller (Ed.), A state-of-the-art report on computational stochastic mechanics, J. Probab. Eng. Mech. 12(4) (1997) 197–321.
[39] G.I. Schuëller (Ed.), Structural reliability software, Struct. Safety—Special Issue 28(1–2) (2006) 1–216. Doi: 10.1016/j.strusafe.2005.03.001.
[40] G. Schuëller (Ed.), Computational stochastic mechanics, Comput. Struct.—Special Issue 85(5–6) (2007) 233–330. Doi: 10.1016/j.compstruc.2006.11.001.
[41] G. Schuëller, H. Jensen, Computational methods in optimization considering uncertainties—an overview, Comput. Methods Appl. Mech. Eng. 198 (1) (2008) 2–13. doi:10.1016/j.cma.2008.05.004.
[42] G.I. Schuëller, H.J. Pradlwarter, Uncertainty analysis of complex structural systems, Int. J. Numer. Methods Eng. 80 (6–7) (2009) 881–913. doi:10.1002/nme.2549.
[43] G.I. Schuëller, H.J. Pradlwarter, P.S. Koutsourelakis, A critical appraisal of reliability estimation procedures for high dimensions, Probab. Eng. Mech. 19 (4) (2004) 463–474. doi:10.1016/j.probengmech.2004.05.004.
[44] G.I. Schuëller, R. Stix, A critical appraisal of methods to determine failure probabilities, J. Struct. Safety 4 (4) (1987) 293–309.
[45] I. Sobol', Global sensitivity indices for nonlinear mathematical modes and their Monte Carlo estimates, Math. Comput. Simul. 55 (2001) 217–280.
[46] C. Soize, A nonparametric model of random uncertainties for reduced matrix models in structural dynamics, Probab. Eng. Mech. 15 (2000) 277–294.
[47] G. Stefanou, The stochastic finite element method: past, present and future, Comput. Methods Appl. Mech. Eng. 198 (9-12) (2009) 1031–1051.
[48] B. Sudret, A. Der Kiureghian, Stochastic Finite Element Methods and Reliability a State-of-the-Art Report, Technical Report, Department of Civil and Environmental Engineering University of California, Berkeley, 2000.
[49] B. Sundaram, B.M. Chapman, L. Bernard, M. Mayo, A. Siddiqui, S. Jones, Sun grid engine package for OSCAR—a Google summer of code 2005 project, in: Proceedings of 20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment HPCS 2006, 2006, p. 41. doi:10.1109/HPCS.2006.42.
[50] S. Tarantola, D. Gatelli, T. Mara, Random balance designs for the estimation of first order global sensitivity indices, Reliab. Eng. Syst. Safety 91 (6) (2006) 717–727. doi:10.1016/j.ress.2005.06.003.
[51] E. Vanmarcke, Random Fields: Analysis and Synthesis, MIT Press, Cambridge, MA, 1998;
Web Edition by Rare Book Services, Princeton University, Princeton NJ, Cambridge, MA, 1998.
[52] F. Yamazaki, M. Shinozuka, G. Dasgupta, Neumann expansion for stochastic finite element analysis, J. Eng. Mech. 114 (8) (1988) 1335–1354.
[53] K. Yang, E. Haik, Design for Six Sigma, McGraw-Hill, 2003.
[54] Structural Use of Concrete—Part 1: Code of Practice for Design and Construction, Technical Report, BS 8110:1985, British Standard Institution, London, 1985.